# Protein Encoder: An Autoencoder-based Ensemble Feature Selection Scheme to Predict Protein Secondary Structure

**Uzma, Usama Manzoor, and Zahid Halim***

***Abstract*—** Proteins play a vital role in the human body as they perform important metabolic tasks. Experimental identification of protein structure is expensive and time consuming. The prediction of protein secondary structure is significant to identify the protein tertiary structure and its folds. The feature subset selection from high dimensional protein primary sequence is a key to improve the accuracy of Protein Secondary Structure Prediction (PSSP). Therefore, it is essential to select the relevant features from high dimensional data to predict the protein secondary structure. This work presents a novel method for the PSSP problem based on a two-phase feature selection technique. The first stage utilizes an unsupervised autoencoder for feature extractions. Whereas, the second stage is an ensemble of three feature selection methods, namely, generic univariate select, recursive feature elimination, and Pearson's correlation. This phase combines multiple feature subsets using mutual information to select the optimum feature subset. For classification, different resultant subset features are used. These include random forest, decision tree, and multilayer perceptron. Two sets of experiments are performed on five datasets for the assessment of proposed work. The proposed solution is compared with three state-of-the-art methods based on Q3 accuracy, Q8 accuracy, and segment overlap score. Obtained results show that the proposed framework performs better in the majority of the cases than the past contributions. The proposed framework achieves Q8 accuracies of 83%, 81%, 80%, 74% and 84% and Q3 accuracies of 89%, 89%, 91%, 78% and 77% on CB6133, CB6133-filtered, CB513, CASP10, and CASP11 datasets, respectively.

***Keywords:*** Protein secondary structure prediction, ensemble methods, autoencoder, feature extraction, amino acids

## 1. Introduction

A gene is a sequence of nucleotides in deoxyribonucleic acid (DNA) or ribonucleic acid (RNA) that acts as its functional unit of heredity (Uzma et al., 2020). Some genes contain instructions for making the functional molecules called proteins. The process from gene to making of a protein inside the cell is complex. It is composed of two stages, transcription and translation. Ribosome reads the bases of mRNA sequences to produce an amino acid chain. The amino acid is glued together by transfer tRNA (tRNA) to assemble the protein by the addition of one amino acid at a time. Fig. 1 shows the central dogma molecular biology. In the living beings, proteins are a type of macromolecule that play a versatile and significant role in all biological processes. Function of proteins in the growth and maintenance of tissues take part in the chemical reactions that occur in our body such as digestion, muscle contraction, blood clotting, and energy generation. Some proteins are hormones which transform information between cells, organs and tissues like proteins and peptides, amines, and steroids. Fibrous proteins such as keratin, collagen, and elastin bring structure, strength, and elasticity to the human body. A few proteins act as an energy provider, maintain blood fluid between tissues, produce antibodies to protect against harmful diseases and carry nutrients into cells, within and outside. The long chain of amino acid is called polypeptide (Flynn et al., 1983) which determines the structure of proteins and the structure dictates the biochemical function. Majority of the proteins are formed by arrangements of same twenty kinds of amino acids, giving rise to the 3D protein conformation and the function of this particular protein entirely depends on its globular structure.

The primary structure of a protein is produced by the linear segment of amino acid residues. Whereas, the protein secondary structure has folded structures that form a polypeptide due to interactions between atoms of the backbone. In molecular biology, protein primary structure sequence is helpful to predict its tertiary structure. However, predicting tertiary structures directly from the primary structure sequences is still a challenge. From the protein primary structure sequence, the secondary structure sequence is predicted and afterwards the secondary structure sequence is used to predict the tertiary structure. Protein secondary structure sequence consists of either three or eight class elements. The three-class secondary structural elements (Torrisi et al., 2018) include: alpha($\alpha$)–helices(H), beta($\beta$)–sheets(E), and coil(C). Whereas, the eight class secondary structural elements are alpha($\alpha$)-helix(H), 310-helix(G), $\pi$-helix(I), $\beta$-bridge(B), $\beta$-sheet(E), Turn (T), Bend (S), and other residues (L).

There are many experimental techniques, like Nuclear Magnetic Resonance (NMR) spectroscopy and Xray crystallography (XRC) that provide high-resolution proteins structure information. However, these techniques are costly, lengthy, and at times unreachable. Moreover, due to the continuing growth of protein databases, the number of unknown protein sequence-structure pairs are constantly increasing. In this situation, cost-effective computational techniques are in demand, which can assist the research community in protein structure prediction. In the past couple of decades, more efforts from computational and experimental perspectives have been made in determining the structure of a protein. However, due to limited development in research, the prediction accuracy is still low (Qian et al.,

* Corresponding author

This work was supported by the GIK Institute graduate program research fund under GA-1 scheme.

*The authors are with the Machine Intelligence Research Group (MInG), Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, 23460. E-mail: uzma.1415@gmail.com,*

1998; Jones et al., 1999). Previously, many statistical methods are proposed for protein secondary structure prediction (Rost et al., 1994; Chen et al., 2006). These models have reportedly round 60% accuracy because they could not find extract features from the primary structure (Kabsch et al., 1983). Nowadays, machine learning techniques are used efficiently to predict the secondary structure. These techniques have been used to predict the secondary structure of a protein including classifiers, like Support Vector Machines (SVM), Artificial Neural Networks (ANN), Random Forests (RF), Convolutional Neural Network (CNN), dynamic Bayesian networks, and ensemble methods. A few of the contributions have also used distance-base machine learning methods for the classification of PSSP, i.e., minimum distance and *k*-nearest neighbor (*k*-NN). These methods' test data is classified using training data and does not require pre-training. Deep Neural Network (DNN) has recently shown potential in various domains, like bioinformatics (Cho et al., 2019;Yu et al., 2021;Araújo et al., 2021), image processing (Pak et al., 2017), speech recognition (Nassif et al., 2019), and signal processing (Gripon et al., 2018). The present work therefore opts to address the PSSP utilizing the DNN in its core.

The work in (Qian et al., 1988) use fully connected Multi-Layer Perceptron (MLP) for protein secondary structure prediction and obtained an accuracy of 64.3%. Jones et al. (1999) use two-stage neural network for protein secondary structure prediction to use PSSM produced by PSI-BLAST and obtain the Q3 accuracy of 79%. The proposal by Pollastri et al. (Pollastri et al., 2002) use a recurrent neural network and profiles eight-class PSSP and obtained Q8 accuracy of 51%. Karypis et al. (2006) used cascaded models based on the pair of binary SVM model and obtained Q3 accuracy of 79.3% and Segment Overlap Score (SOV) of 78.7%. Zhon et al., (2007) use parallelize Denoeux Belief Neural Network (DBNN) to achieve a speedup of 4-4.9 and obtained Q3 accuracy of 72%. Aydin et al. (2007) developed two search algorithms, based on N-best score for Hidden Semi Markov Model (HSMM) and obtained the Q3 accuracy of 64%. Yao et al. (2008) used dynamic Bayesian networks and obtained the Q3 accuracy of 77.5%. Sonderby et al. (2014) use BRNN with LSTM for protein secondary structure prediction and achieved the Q8 accuracy of 67%. Li et al. (2016) apply cascaded convolutional neural networks and recurrent neural network for PSSP and achieved a Q8 accuracy of 69.7%. Wang et al. (2016) utilize SVM with PSSM profiles and achieves the accuracy of 76.11%. Busia et al. (2017) apply deep convolutional neural networks and attain the Q8 accuracy of 71.4%. Guo et al. (2018) use recurrent neural network and 2D convolutional neural networks to attain Q8 accuracy of 70%. Ma et al. (2018) utilize data partition and semi-random subspace method for PSSP and achieved the accuracy of 84.5% on CB513 dataset. Guo et al. (2019) apply deep asymmetric convolutional long short-term memory neural models for PSSP and attained the Q8 accuracy of 75%. Kumar et al. (2020) use deep learning framework on hybrid profile-based features and obtained the Q8 accuracy of 75.8% and 73.5% on CB513 and CB6133 datasets. Aydin et al. (2018) use dimension reduction

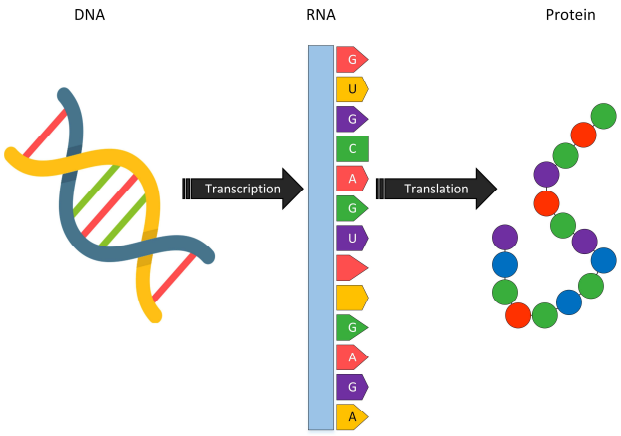


Fig. 1. Central dogma molecular biology.

techniques for PSSP and opted for two feature selection methods. They use support vector machine in second stage for the classification and achieved the Q3 accuracy of 83.05% and SOV score of 80.32 on CB513 dataset. Kathuria et al. (2018) used the RF classifier to predict the unknown proteins. Hu et al. (2020) used the RF classifier to predict super-secondary structure in proteins. Random forest classifier is used to improve the model accuracy with minimum classification error. It can effectively avoid the overfitting phenomenon by incorporating appropriate randomness and works well on most regression and classification problems. An RF algorithm has been used in protein-RNA binding sites, enzyme catalyst residues, helical domain linker, and oligomer status of coiled helical regions. This enables better results (Song et al., 2018; Okun et al., 2007; Jia et al., 2011; Richa et al., 2017; Liu et al., 2010). Yavuz et al. (2018) use MLP classifier for prediction of protein secondary structure. Dencelin et al. (2016) use multilayer perceptron for the classification of protein secondary structure. MLP is a faster machine learning model. It uses feed-forward and backpropagation. Selbig et al. (1999) use Decision Tree (DT) for the consensus secondary structure of protein. Dowe et al. (1993) use DT for graph explanation of PSSP. Decision tree is used to improve the accuracy. It requires consideration of all possible outcomes of a decision and draws conclusions by overcoming the overfitting issue. Amino acids residues form a high dimensional data. The large feature vector contains redundant, unimportant, and missing values. There is some work published in the past to address the challenge. For example, Kumar at al. (2020) design a framework based on the combination of CNN and BRNN for the extraction of local and long-term interconnection between amino acids. Li et al. (2016) use the CNN to extract the local context and the Bidirectional Gated Recurrent Unit (BGRU) to extract the global context. Guo et al. (2018) use the 2-D convolutional neural network (2C) for local amino acid interaction. They use Bidirectional Recurring Neural Networks (BRNNs) to manage the global interaction between amino acid residues. These frameworks are complex models based on the combination of deep neural network. Therefore, we need to design a simple model that accurately predicts a protein's secondary structure. Therefore, the current solution is designed on the basis of unsupervised deep learning with

feature selection methods to extract the set of meaningful features. The proposed work is a novel deep learning method referred to as *Protein Encoder*. It uses the unsupervised deep learning method (autoencoder) with feature selection techniques for reducing the dimension of amino acid residues and assist in selecting relevant features. The auto encoder has the capability of learning the non-linear relationship between features. It finds a good representation of input data in low dimension by focusing on significant features and ignoring redundant and noisy data. The features are extracted by converting the high-dimensional amino acid residues to small-dimensional. Additionally, to select the most relevant features that play an important role in classifying proteins into different groups, three feature selection methods are combined. The subset of features generated by different selection methods is aggregated using the aggregation function. An ensemble feature selection method is used, as each feature selection method uses different criteria to select the features. Therefore, it is inappropriate to use a single method for selecting features. As a result, the proposed solution aggregates three feature selection method, i.e., if one method ignores the important feature the other one selects it. The generated feature subset contains important information for predicating protein structures. As a result, data based on the selected subset of features is provided to the classifiers to group the protein structure in order to analyze the protein functions.

### 1.1. Problem Statement

Protein Data Bank (PDB) is a global repository of data about the 3D structures of large proteins and nucleic acids (Burley et al., 2017). PDB identifies the expression level of hundreds of proteins simultaneously. The huge protein primary structure data makes it challenging to process. This data has noise, unnecessary and irrelevant items that makes it difficult to process. In the literature, models commonly used for reducing the dimensions of the data are polypeptide composition, Amino Acid Composition (AAC), functional domain composition, PSI-BLAST profiles, physicochemical feature, and function annotation information. When information about protein properties is extracted, it often contains significant redundancy, leading to unsatisfactory levels of recognition for structural classes of proteins. Protein dataset also contain noise. For classification, data is critical for its accuracy and efficiency. Too many features can increase training time and cause overfitting, which reduces the accuracy on unknown data (Uzma et al., 2021). Also, noisy features can cause distortion in training. Whereas, a few features may not be sufficient for satisfactory training and causes under fitting. Therefore, a suitable and adequate number of features must be used to train the model. To solve the abovementioned problems, dimensionality reduction techniques, such as feature selection and feature extraction methods can be used (Han et al., 2011). The main difference between these two techniques is that when feature selection technique is applied, some of the features are selected without any change in the data. However, the feature extraction reduces the size of the data but creates new feature set. These techniques not only reduce the size of large data by selecting the important features but also improves the classifiers' performance, consumes less resources and speedups the model.

### 1.2. Key Contributions and Novelty

The proposed solution here is a novel three-phase approach to protein residue prediction. For the selection of relevant features, two-stage techniques are employed. The first phase use unsupervised deep learning method called autoencoder for extracting features. Protein data is given as an input to an autoencoder that is converted to low-dimensional data at the code layer. Once the network is trained, the code layer data is used as input for the second phase of the proposed approach. Afterwards, the three feature selection methods are applied to the code layer, then the resulting subset of features is aggregated using Mutual Information (MI). Finally, that samples are classified based on the selected features subset by using various classifiers.

This work presents a novel approach to predict protein secondary structure. The intent of the proposed methodology is to select relevant amino acid attributes for protein domain classifications. The method selects the optimal subset of features from high-dimensional protein residues data. For the accurate predication of protein's domain, the selection of most relevant features play an important role. Filter method of feature selection is used to reduce the feature dimensions in the current framework. Following are the key contributions of this work.

- Autoencoder-based reduction of features
- Fusion of three feature selection methods for optimum feature subset selection
- Ensemble of multiple classifiers, namely, random forest, decision tree, and multi-layer perceptron for predicting the protein domain based on the subset of the selected optimum features.

The novelty of the present work is that it uses three stage approach for PSSP based on deep learning methods. In the first two stages of the proposed model, dimension reduction techniques are used. First phase is an unsupervised autoencoder module for the dimension reduction and feature extraction. The output data of the autoencoder's code layer is divided into training and testing sets. Second stage of the present work has an aggregation of three feature selection technique. Different features' subsets are obtained by the three feature selection methods and then aggregate of different subsets is taken to obtain TopN features' data. These techniques are applied on the training dataset. For aggregation, ensemble feature selection using MI (Hoque et al., 2018) algorithm is used. It directly selects common features that are selected by all feature selection methods, otherwise selection of the features is done on the basis of feature-to-feature MI and feature class MI. After this stage, reduced data with optimal features is passed to the classification module.

The rest of the paper is organized as follows. Section 2 discuss related work. Section 3 lists the biological material. Section 4 presents the proposed method. Section 5 lists the conducted experiments and outcomes. Finally, Section 6 provide concluding remarks about the proposed work.

## 2. Related Work

This section covers the past works done using feature selection techniques, machine learning, and deep learning methods to address classification challenges in the protein data. The PDB simultaneously develops structure of thousands of proteins. It organizes protein data in a matrix form, with rows representing the amino acids and columns representing the features. The combination of row and columns represent the protein structures profile and each entry represent structure of the given protein (Ding et al., 2014). With the passage of time, the need of protein structures' data is increasing. It is a technique that extracts important biological information that assists the domain of medicine and biotechnology (Gowthaman et al., 2021). The analysis of many datasets of protein structures is challenging. Thus, it is necessary to build a tool to obtain and analyze meaningful biologically information from enormous protein structure datasets. For the analysis of high-dimension data, classification is a useful learning technique (Al-Obeidat et al., 2020). It is challenging to process the protein structure data because of large number of proteins. Each protein has several states and some of the data is redundant. Furthermore, the dataset generated by Protein Data Bank (PDB) is noisy and redundant.

### 2.1. Statistical Methods

The earliest methods used for protein secondary structure prediction are statistical approaches. These models were based on statistical information of single amino acid and limited to a few proteins of known structures. Although these methods are not advanced, however, they are still used as an initial step of protein secondary structure problem. Examples of such methods include: Chou-Fasman algorithm (Chou et al., 1974) and Garnier-Osguthorpe-Robson algorithm (Garnier et al., 1978). The Chou-Fasman algorithm is based on the frequencies of amino acid. Frequency of helices, sheets, and coils is determined by the X-ray crystallography of known protein structures. Based on these frequencies the probability parameters are set that derived the appearance of amino acid in different secondary structures. Probability parameters predict that the given amino acid sequence is helix, stand or coil in a protein. The prediction accuracy of this method is between 50% to 60% (Kabsch et al., 1983). The Garnier-Osguthorpe-Robson (GOR) algorithm is a theory-based method. It is also based on probability parameters, like Chou-Fasman algorithm. Probability parameters are determined by the X-ray crystallography of known protein structures. GOR method not only considers the probability parameters of specific secondary structure, but also takes into account the conditional probability of immediate neighboring structures that already formed the same structure. The prediction accuracy of this method is around 57%.

### 2.2. Machine Learning Methods

Multiple machine learning methods have been used to predict the secondary structure, including ANN, SVM, dynamic Bayesian networks, RF, and ensemble techniques (Halim et al., 2020). Salamov et al. (1995) work on ANN and *k*-NN to achieve 72.2% Q3 accuracy. Jones et al. (1999), use neural networks on PSSM calculated by PSIBLAST algorithm and attain 76.5% - 78.3% Q3 accuracy. Yao et al. attain 78.1% Q3 accuracy by a method of dynamic Bayesian network and neural network (2008). The *k*-NN and minimum distance (using the distance formulas, like Minkowski or Euclid distances) are used to classify the data of protein structures. These algorithms do not need pre-training data. Ghosh et al. use *k*-NN, minimum distance, and fuzzy *k*-NN algorithm on a protein structures dataset and they compared these methods with multilayer neural networks. The authors show that these methods work better and attain higher accuracy than the multilayer neural networks (Fayech et al., 2013). Hidden Markov Model (HMM) is also used to predict protein secondary structure. It estimates the future behavior based on existing data. It is commonly used as a classifier in many fields such as data mining, image processing, bioinformatics and pattern recognition, to name a few. Aydin et al. (2006), use extended hidden semi-markov model to predict secondary structure for single sequence and attained 67.9% Q3 accuracy. Another commonly used method to predict the protein secondary structure is the SVM. SVM usually utilize linear hyperplane for data distribution. Huang et al. (2013), apply SVM on a dataset that is generated using PSSM values and four physicochemical features and achieve 79.5% Q3 accuracy.

The aim of protein secondary structure prediction is to assign secondary structural elements such as alpha helix, beta sheet and coil, for each amino acid. Therefore, the number of samples in the datasets will be equal to the number of amino acids, which can be large. In this case, it is important to speed up the learning algorithm. Fully connected neural network is used for large dataset. Huang at al. (2006), first propose the fully connected neural network. Wang et al. (2008), use

Table 1
Summary of Past Works

| Authors | Proposed Model | Accuracy |
|---|---|---|
| (Sonderby et al., 2014) | BRNN with LSTM | Q8-67% |
| (Li et al., 2016) | Cascaded convolutional neural networks and recurrent neural network | Q8-84.5% |
| (WangIn et al., 2016) | Support vector machine (SVM) with PSSM profiles | Q8-76.11% |
| (Busia et al., 2017) | Deep convolutional neural networks | Q8-71.4% |
| (Guo et al., 2018) | RNN and 2D CNN | Q8-70% |
| (Ma et al., 2018) | Utilized semi-random subspace method and data partition | Q8-84.5% |
| (Guo et al., 2019) | deep asymmetric convolutional LSTM neural models | Q8-75% |
| (Kumar et al., 2020) | Used deep learning on hybrid profile-based features | Q8-75.8% |

Table 2
Three and Eight State Secondary Structure

| State | Name | Single Letter Code |
|---|---|---|
| 3-state | α-Helix | H |
| | B-Strand | E |
| | Coil | C |
| 8-state | residue in isolated β-bridge | B |
| | Extended strand | E |
| | 3-10 helix | G |
| | α-helix | H |
| | π-helix | I |
| | Hydrogen bonded turn | T |
| | Bend | S |
| | Loop or any other residues | L |

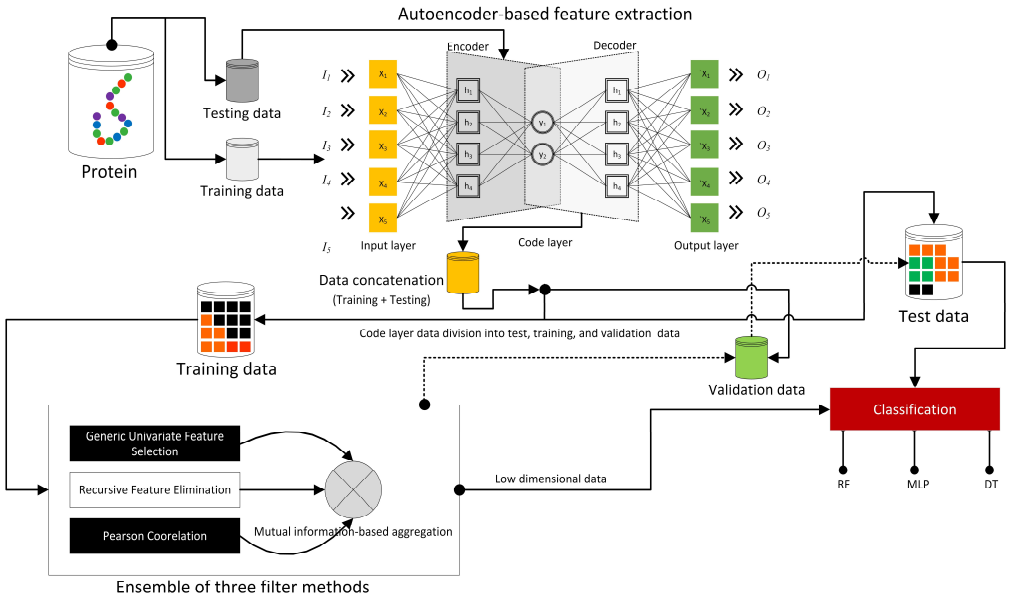



Fig. 2. Overall working of the proposed solution.

extreme learning machine on protein datasets and achieve 74.7% Q3 accuracy. At times, the classification algorithms do similar errors when compared to each other, in this case it is probable to make errors belonging to a specific class. To avoid such kinds of errors, ensemble methods are used, which utilize mathematical/statistical techniques to combine two or more classification algorithms. Bouziane et al. (2015), combine ANN and SVM on CB513 dataset. The model achieves 78.50% Q3 accuracy. Li et al. (2017), use Principal Component Analysis (PCA) on a dataset and attain 86.7% Q3 accuracy through SVM. Fayech et al. (2013), apply data mining for protein secondary structure prediction and achieve 78.2% Q3 accuracy. Shuai-yan et al. (2017) propose a radical group encoding method for PSSP. The purpose of their encoding scheme is to encode the 20 amino acids of protein. All amino acids are represented by coding with the information of stable structure of those atoms that exist in the amino acids protein. Experiments are performed on CB513 and 25PDB datasets with variable window length. For classification, SVM and Bayes classifiers are used in their model. This method is compared with the quadrature encoding and archives 1.2% better accuracy. Liu et al. (2017), propose a two dimensional deep convolutional neural networks for the large data of proteins. It is composed of six convolutional layers and five max-pooling layers. They use six benchmark datasets. Wang et al. (2019), use ensemble method, where length and width of 2D data is passed to the time dimension of two different LSTM models and a third LSTM model is designed to combine the results of first two models. Holley et al. (1989), used the feed-forward neural network for the predication protein secondary structure. The method was evaluated on 64 proteins having the first 48 proteins are used for training and the next 14 is used for testing, the accuracy is 79%.

### 2.3. Limitations of the Past Work Addressed

In the previous literature, most of the techniques are designed and applied directly to the proteins datasets without any feature selection layer. A few works use feature selection of protein datasets which speeds up the computational model (Li et al., 2017). Different feature selection methods give different feature subsets; therefore, selecting the optimal feature subset is a challenging task. The work in (Cho et al., 2019) use dimension reduction techniques for protein secondary structure production and opts for different feature selection

methods, i.e., chi-square, gain ratio, basic component analysis, the minimum redundancy maximum relevance algorithm, correlation-based feature selection, and information gain, and then apply SVM for the classification of each feature subset.

The work in (Li et al., 2017) use only one method for feature selection. Afterwards, the top ranked features are used for classification. As a single feature selection method does not give an optimal subset of features, so different feature selection methods that provide varying feature subsets needs to be evaluated. Hence, the present work uses an ensemble of three feature selection methods. If important features are ignored by one feature selection technique, there is a possibility that the other method selects it. In the previous work the models take a lot of time and computational resources due to large number of data (Kumar et al., 2020). The current framework uses dimension reduction and feature selection techniques for reduced the size of large data by selecting the important features that improves classifiers' performance, consume less resources and speedups the model. For this purpose, autoencoder is used that converts high dimensional data into low dimensions and extracts relevant features. Table 1 lists the summary of past works.

**3. Biological Material**

This section explains the required biological material. The protein primary structures data contain 20 amino acids, denoted by A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y. "A" denotes Alanine, "C" denotes Cysteine, "D" denotes Aspartic Acid, "E" denotes Glutamic acid, and so on. "X" is used to denotes unknown amino acid. Tertiary structure of protein is predicted by the protein secondary structure. Protein profile is a powerful representation of primary protein structure rather than representing each amino acid separately. These amino acids are used to consider evolutionary regions and are utilized to model protein groups and domains. They are formed by conversing of multiple sequence alignments into Position-Specific Scoring Matrix (PSSMs). The frequency of each amino acid at the given position is calculated and then according to these frequencies the amino acids position in the alignment are scored (Iqbal et al., 2021). Different combination of amino acids makes the protein's polypeptide chain. A protein's polypeptide chain normally contains round 200-300 amino acids, but it can consist of less or more amino acids. In the used datasets here, the average protein's polypeptide chain consists of 208 amino acids. The structural state of amino acid residues in the protein is determined by the protein secondary structure. The coiled-up shape of protein is formed by $\alpha$-helix, zigzag shape is formed by β-stand. The protein secondary structure is important because it has many chemical properties of protein and is used for predicting the tertiary structure of protein. When predicting protein's secondary structure, one differentiates between 3-state secondary structure and 8-state secondary structure as shown in Table 2.

Five benchmark datasets (Zhong et al., 2007) are utilized for the evaluation of the proposed work. These include: CullPdb6133, CullPdb6133-filtered, Cuff, Barton's 513 (CB513), CASP10, and CASP 11. These are publicly available datasets for PSSP. CullPdb6133, CullPdb6133-filtered and Cuff and Barton's 513 (CB513) are available in numpy format with $N$ protein x $k$ features. CASP10 and CASP 11 are available in Hierarchical Data Format 5 (HDF5). The CullPdb6133 dataset contains the 6133 protein sequences with 39900 features of each protein. These proteins can be reshaped into 6133 proteins x 700residues x 57 features. Cuff and Barton's 513 (CB513) dataset contains 513 protein sequences with 39900 features of each protein. CASP10 dataset contains 128 protein sequences and CASP11 dataset contains 105 protein sequences. For data consistency here, 700 x 57 matrix of amino acid chains are formed. Where, the digit 700 represent the peptide chain of protein and 57 represent the feature set of amino acid. No sequence (zero padding) in vector is applied when the end of chain is reached. Among the 57 features, 22 features (i.e., from 0-21) represent the protein primary structure (20 amino acid residues, 1 unknown amino acid residues, 1 none sequence/padding). Nine features (i.e., feature number 22-30) represent the secondary structure labels (8 possible states, 1 no sequence or padding). Two features (i.e., from 31-32) represent the C- and N- terminals. Two features (i.e., feature number 33-34) represent the relative and absolute solvent accessibility. Twenty-two features (i.e., 35-56) represent the protein sequence profile. The protein sequence profile is used for protein primary structure prediction. The proposed method takes a sequence of amino acid as the input. Initially, the amino acid sequence (700 x 22) and labels (700 x 9) are extracted from the dataset. Afterward, the no sequence/padding is removed from the amino acid sequence (700 x 21) and labels (700 x 8). The sequence feature vector is a sparse one-hot vector, i.e., only one of its eight elements is none-zero (1), while the sequence vector has a dense representation. To avoid inconsistency of feature representation, we transform the sparse one-hot vector into the dense vector by an embedding operation (Guo et al., 2018). After that, concatenation of the 6133 amino acid sequence matrices is performed that combines them into one matrix (4293100 x 21). This matrix is then passed to the autoencoder for feature reduction. From the code layer of autoencoder, a compressed matrix (4293100 x 18) is obtained. Before data passing to the autoencoder, it is divided into the training set (70%) and testing set (30%) randomly. The sequence of amino acid residue is passed as input to the autoencoder. After passing through the input layer, the data is passed to the hidden layer and then to the code layer. The dimension of hidden layer is 19 and code layer is 17. The output data of code layer, i.e., the 17-dimensional feature map, is further used in the next step. However, concatenation of the training and test data is performed first. In the next step, three classifiers are applied on the 17 code layer data. For each data and classifier, 10 experiments are performed, using all five datasets (i.e., CB6133, CB6133 filtered, CB513, CASP10 and CASP11). Before training the data is split into train, validation,

**Input:** Data, Features of best set (N), threshold (μ)
**Output:** S, denotes the selected feature subset

```
1.  FS1, FS2, FS3 are selected subset of features using three different feature selection methods
2.  Initialize S ← φ and a counter i ← 1
3.  while i ≤ S do
4.       if ({FS1[i]} == {FS2[i]} == {FS3[i]}) then
5.       S ← S ∪ {FS1[i]}
6.       else
7.       Calculate feature-class Mutual Information ({FSn[i]}, C), ∀j ∈ [1, 2, 3]
8.       end if
9.  Select those feature f that have maximum feature-class Mutual Information
10.     if (S==NULL) then
11.     S ← S ∪ {f}
12.     else
13.     Calculate feature-to-feature MI (f, fs), for f with all the selected feature fs ∈ S
14.         if Calculated information is less than μ for all selected features in S then
            S ← S ∪ {f}

15.         end if
16.     end if
17.         i←i+1
18. end while loop
```

**Algorithm 1.** Ensemble feature selection using mutual information.

and test set. CB6133 dataset consist of 6133 proteins. In this dataset 5600 proteins from 0 to 5600 are used for training, 272 proteins from 5605 to 5877 are used for test and 256 proteins from 5877 to 6133 are used for validation. CB513, CASP10 and CASP11 data are testing data. There are 25% similarity between the CB6133 and CB513 dataset. The filtered version of CB6133 is generated by removing the similarity. CB6133-filtered dataset consists of 5534 proteins (Guo et al., 2018). It is used for training and the other three test datasets are used for testing. Furthermore, the CB6133-filtered dataset split into train, validation, and test set of 5060, 244 and 230 proteins. Then CB6133-filtered dataset is used to train and test the proposed model.

## 4. Proposed Solution

This section presents the proposed solution to the protein secondary structure prediction problem. It is characterized into three main stages. The challenge here is to analyze protein dataset that has very large number of samples. This work presents three-stage method for this, which include autoencoder-based feature extraction, ensemble filter methods for relevant feature selection, and finally classification. In the first stage features are extracted using an autoencoder. The autoencoder is utilized to find better representation of the input data. From the input data, the unrelated features are removed through the autoencoder. The train data is passed to the next stage. In the second stage the ensemble-based features selection method is used for removing the redundant, irrelevant, and noisy data. The ensemble method provides an optimal set of features over the single feature selection method. Therefore, three methods of feature selection, namely, generic univariate select, recursive feature elimination, and Pearson correlation coefficient are used. From each feature selection technique topN ranked attributes (where topN is set to 5, 10, and 15 in the experiments) are selected. Next, the feature subset generated by these methods are aggregate using MI (Hoque et al., 2018) to get an optimum feature subset. Similar features are directly selected from the train dataset.

Afterwards, based on the selected feature subset, protein domain is classified by three classifiers, i.e., multi-layer perceptron (MLP), random forest, and decision tree. Overall working of the proposed framework is shown in Fig. 2. Following sections explain the individual components of the proposed solution.

### 4.1. Autoencoder

Autoencoder is an unsupervised learning technique based on the traditional ANN for the task of efficient data representation. This work constructs an architecture of the ANN such that it imposes a bottleneck in the network which enable a compressed knowledge representation of the input data (Uzma et al., 2020). Autoencoder mainly consists of three parts: an encoder maps, input to the code layer, and the decoder part that reconstructs the original input from the compressed data. In the model, the reconstruction loss measures the difference between the input and output data. If the input features are independent to each other, the encoding and decoding would be challenging. However, if some sort of correlation between the data exists, the encoding and decoding would be convenient. Autoencoder is an unsupervised deep neural network. The proposed solution use an autoencoder for the dimension reduction and feature selection. Classification of proteins is an important task for identifying the structures/functions of the unknown protein sequences. The accurate representation of amino acid residues during the extraction of features is one of the main problems related to the classification of proteins. In the proposed work, the selected dataset is preprocessed prior to applying the training and testing phase. Each protein sequence is represented by a feature vector (Fv). The data has many redundant and irrelevant features providing no information about the protein sequences. This influences performance and runtime of the classification algorithms. Therefore, the proposed solution aims to remove features that do not contribute towards the representation of protein sequences. The proposed work first alters the original feature representation. Next, various feature selection methods are used to select the most appropriate subset of features. Therefore the present work is based on two phase feature selection technique, including: autoencoder-based feature extraction and an ensemble of three feature selection methods for relevant feature selection.

### 4.2. Feature Selection

Feature selection is the method of selecting the relevant feature subset. These selected features are then used in learning model construction. The aim of feature selection is to remove redundant or irrelevant features from the data. There are several reasons of using feature selection techniques, like generalization of models (James et al., 2013), reduce training times, reduce computational resources, and reducing overfitting (Bermingham et al., 2015), to name a few. This work utilizes three feature selection techniques. Details of these are listed in the following.

#### *4.2.1. Generic Univariate Select*

Univariate feature selection is a filter-based method that analyzes each feature independently to determine the relationship of the feature with the target variable. Based on univariate statistical tests, univariate feature selection technique selects the best feature subset. According to certain criteria it ranks each feature. Generic univariate select is a sklearn feature selection tool that work on scoring function and allow to select features from a dataset. It supports selecting features in one of a few various configurations; k for selecting a specific number of features and percentile for select a percentage of the total number of features.

#### *4.2.2. Pearson Correlation*

Pearson correlation is a commonly used measure in machine learning. It is a filter method for feature selection and used for numerical input and output. It is a popular method of determining the relationship between variables of interest because it is based on the notion of covariance. It provides information regarding magnitude of the relationship, or correlation and direction of the association. It evaluates the statistical relationship between two data variables. Pearson correlation value ranges between -1 and 1. Computation of Pearson correlation coefficient is shown in Eq. (1).

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2+\sum_{i=1}^{n}(y_i-\bar{y})^2}} \quad (1)$$

Where, $r$ denotes the correlation coefficient, $x_i$ denotes the values of the sample x-variable, $\bar{x}$ represent the x-variable mean, $y_i$ denotes the values of the sample x-variable, and $\bar{y}$ represent the y-variable mean.

A value of 1 or closer to it indicate a positive correlation between two variables. It shows that there is a direct relation between two variables. A value of -1 or closer indicate a negative correlation between two variables. It shows that there is an inverse relation between two variables. A value of zero (or near to it) indicate no correlation between two variables.

#### *4.2.3. Recursive Feature Elimination*

Recursive Feature Elimination (RFE) is a commonly used feature selection algorithm. It is helpful for selecting those features in a dataset that are more important in predicting the target variable. RFE is a wrapper feature selection algorithm but internally it also uses filter-based feature selection. Different machine learning algorithms are used in the core of the method and are wrapped by REF for features selecting. RFE searches for a subset of features, starting with all features in the training dataset, and manages to remove features until the desired number is retained.

#### *4.2.4. Aggregation Function*

The feature subsets generated by the ensemble methods are combined using the aggregation function. The proposed solution combines the feature subset through MI based aggregation function called Ensemble Feature Selection using Mutual Information (EFS-MI) (Hoque et al., 2018). The EFS-MI method use greedy search approach for combing the subsets of features that are selected by different feature selection methods. If all feature selectors choose a common feature, then that is put into the optimal subset without using greedy search method, otherwise the EFS-MI computes the feature-to-feature mutual information and feature-class mutual information and selects a feature that has minimum feature-to-feature MI and maximum feature-class MI. The biasness produced by each feature-selection method is removed by this method. Feature-to-feature MI removes the redundancy among the selected features and feature–class MI selects relevant features. Overall working of the EFS-MI is listed in Algorithm 1.

### 4.3. Classification Model

Classification is an important part of machine learning. Classification model aims to get some decision from the input data given for training. It is a supervised technique that use the labeled training dataset to identify the class of the new observation. Once the model is trained on the train data, it then predicts the target class. The present work uses following three classifiers.

#### *4.3.1. Multilayer perceptron*

A multilayer perceptron (MLP) is a supervised classification method based on artificial neural network. It is created using more than one perceptron (neuron). Major part of MLP is an input layer, an output layer and one or more hidden layers. Input layer receive the data, output layer makes prediction about the input data and hidden layers performs computation. Any continuous function can approximate with MLP of one hidden layer. MLP is feedforward network used to forward pass and backward pass the information. In forward pass, the data initially pass through the input layer and then pass through the hidden layers and finally pass-through output layer. The result of the output layer is compared with the ground truth. Backpropagation is used in the backward pass. In backpropagation, the error function value is computed with respect to weights of MLP and is passed backward to the network. The parameters are adjusted to minimize the error. MLP continues the backpropagation process until the model converges.

#### *4.3.2. Random Forest (RF)*

Random forest is a supervised classification method. As the name indicate, it creates a group of decision trees, mostly trained with the bagging method. The common concept of the

bagging method is that it randomly creates decision trees and combines them together to improve the whole result (Mao et al., 2012). Random forest can be used for both regression and classification problems. In classification, instead of searching for the most important feature random forest search the random subset of features. It generates the diversity in RF, as a result the model predicts better. Therefore, in RF, only the random subset of features is accounted for by the node sharing algorithm. One can even create a tree more random by using a random threshold for each function rather than finding the best threshold (as normal decision trees does).

#### *4.3.3. Decision Tree (DT)*

Decision tree is a supervised learning method in machine learning. It is mostly used for classification problems but also utilized in regression tasks. As the name indicate, it creates a tree, that start with root node, which grows on further internal nodes and ends with leaf nodes. Root node has one incoming edge and zero or more outgoing edges, internal nodes have one or more incoming edges and two or more outgoing edges, Leaf nodes have one or more incoming edge and zero outgoing edges. Root node and internal nodes are called decision nodes because these nodes are used to make any decision whereas leaf nodes represent those decisions. A decision tree basically asks a question, it further split the tree into subtrees based on the answer to the question, which is either yes or no (Wu et al., 2008).

Amino acids residues form a huge dimensional data. The large feature vector contains redundant, unimportant, and missing values. There is some work published in the past to address the challenge. For example, Kumar at al. (2020) design a framework based on the combination of CNN and BRNN for the extraction of local and long-term interconnection between amino acids. Li et al. (2016) use the CNN to extract the local context and the Bidirectional Gated Recurrent Unit (BGRU) to extract the global context. Guo et al. (2018) use the 2-D convolutional neural network (2C) for local amino acid interaction. They use Bidirectional Recurring Neural Networks (BRNNs) to manage the global interaction between amino acid residues. These frameworks are complex models based on the combination of deep neural network. Furthermore, it is necessary to design a simple model that addresses the challenges of predicating the secondary structure of protein in an efficient manner. Therefore, the current solution is designed on the basis of unsupervised deep learning with feature selection methods to extract the set of meaningful features. The proposed work is a novel deep learning method referred to as Protein Encoder. It uses the unsupervised deep learning method (autoencoder) with feature selection techniques for reducing the dimension of amino acid residues and assist in selecting relevant features. The auto encoder has the capability of learning the non-linear relationship between features. It finds a good representation of input data in low dimension by focusing on significant features and ignoring redundant and noisy data. The features are extracted by converting the high-dimensional amino acid residues to small-dimensional. Additionally, to select the most relevant features that play an important role in classifying proteins into different groups, three feature selection methods are combined. The subset of features generated by different selection methods is aggregated using the aggregation function. An ensemble feature selection method is used, as each feature selection method uses different criteria to select the features. Therefore, it is inappropriate to use a single method for selecting features. As a result, the proposed solution aggregates three feature selection method, i.e., if one method ignores the important feature the other one selects it. The generated feature subset contains important information for predicating protein structures. As a result, data based on the selected subset of features is provided to the classifiers to group the protein structure in order to analyze the protein functions.

## 5. Experiments and Results

This section presents the experiments preformed to evaluate the proposed framework. Five benchmark protein datasets are used for this. The proposed solution is compared with three closely related state-of-the-art methods, i.e., Yanbu Guo et al. (2018), Li et al. (2016), and Kumar et al. (2020). Two types of experiments are performed for evaluating the framework: (a) evaluation of the TopN selected features using different classifiers and (b) evaluation of the extracted features using multiple classifiers. For experiment: (a) 10 experiments are performed for each of the TopN (i.e., by setting TopN to 5, 10, and 15 respectively) subset data with each classifier, and (b) 10 experiments are performed for the extracted data with each classifier.

### 5.1. Evaluation Metrics

Performance of the proposed solution and three competing methods is evaluated using six metrics, namely, SOV (1994), precision, recall, f1-score, Q3 for 3-class PSSP, and Q8 for 8-class PSSP.

#### *5.1.1. Segment Overlap Score*

The SOV is a commonly used metric in the domain of bioinformatics. It is used to compare two sequences of letters in which a continuous segment is important. The advantage of SOV is that it can consider the size of the continuous overlapping segments and provide additional tolerance for longer continuous overlapping segments, rather than just assessing the percentage of individual positions that overlap, as Q3 does (Liu et al., 2018). The computation of SOV score is shown in Eq. (2).

$$SOV = \frac{1}{N} \times \sum_{i} \sum_{S(i)} \frac{minov\left(s_1^i,\ s_2^i\right) + \delta(s_1^i,\ s_2^i)}{maxov(s_1^i,\ s_2^i)} \times \left|s_1^i\right| \times 100 \quad (2)$$

Where, $N$ is the number of amino acid residues in a protein. In the case of 3-class PSSP, $i$ is the element of {H, E, C} and in 8-class PSSP, $i$ is the element of {B, E, L, G, H, I, S, T}, $S_1^i$ denotes the actual protein secondary structure and $S_2^i$ denotes the predicted protein secondary structure. The pair $\left(s_1^i,\ s_2^i\right)$ denotes the overlapping between actual protein secondary structure and predicted protein secondary structure, $minov\left(s_1^i,\ s_2^i\right)$ shows the overlapping length of two segment

Table 3
Performance of the Proposed Solution with Random Forest Classifier on 8 class data and 3 class data

| Data | Features | Datasets | Precision | Recall | F1-score | Q8/3 Accuracy | SOV8/3 |
|---|---|---|---|---|---|---|---|
| Eight class data | Five feature data (F5) | CB6133 | 0.7993 | 0.8012 | 0.8002 | 0.7881 | 0.7776 |
| | | Cb6133_filtered | 0.7945 | 0.7958 | 0.7951 | 0.7553 | 0.7323 |
| | | CB513 | 0.7993 | 0.8064 | 0.8028 | 0.7776 | 0.7465 |
| | | CAPS10 | 0.7236 | 0.7308 | 0.7272 | 0.7107 | 0.7038 |
| | | CAPS11 | 0.7567 | 0.7566 | 0.7566 | 0.7427 | 0.7268 |
| | Ten feature data (F10) | CB6133 | 0.8293 | **0.8313** | 0.8303 | **0.8261** | **0.8116** |
| | | Cb6133_filtered | **0.8245** | **0.8265** | **0.8255** | 0.8053 | **0.7873** |
| | | CB513 | **0.8113** | **0.8163** | **0.8138** | **0.8157** | **0.8012** |
| | | CAPS10 | 0.7304 | 0.7301 | 0.7302 | **0.7405** | **0.7234** |
| | | CAPS11 | 0.7536 | 0.7668 | 0.7601 | 0.7628 | 0.7366 |
| | Fifteen feature data (F15) | CB6133 | **0.8333** | 0.8283 | **0.8308** | 0.8176 | 0.8012 |
| | | Cb6133_filtered | 0.8212 | 0.8261 | 0.8236 | **0.8074** | 0.7805 |
| | | CB513 | 0.8003 | 0.7966 | 0.7984 | 0.7801 | 0.7643 |
| | | CAPS10 | **0.7464** | **0.7511** | **0.7487** | 0.7295 | 0.7106 |
| | | CAPS11 | **0.7737** | **0.7689** | **0.7713** | **0.7701** | **0.7568** |
| Three class data | Five feature data (F5) | CB6133 | 0.7321 | 0.8102 | 0.7692 | 0.8256 | 0.8052 |
| | | Cb6133_filtered | 0.7544 | 0.7832 | 0.7685 | 0.8265 | 0.8094 |
| | | CB513 | 0.7873 | **0.8407** | 0.8131 | 0.8211 | 0.8016 |
| | | CAPS10 | 0.7612 | **0.7812** | **0.7711** | 0.7506 | 0.7346 |
| | | CAPS11 | **0.7734** | 0.7663 | 0.7698 | 0.7709 | 0.7507 |
| | Ten feature data (F10) | CB6133 | 0.8311 | 0.8133 | 0.8221 | 0.8416 | 0.8222 |
| | | Cb6133_filtered | **0.8523** | **0.8654** | **0.8588** | **0.8367** | **0.8197** |
| | | CB513 | 0.8312 | 0.8003 | **0.8155** | **0.8317** | 0.8112 |
| | | CAPS10 | 0.7413 | 0.7132 | 0.7270 | 0.7519 | 0.7443 |
| | | CAPS11 | 0.7633 | 0.7812 | **0.7721** | 0.7907 | 0.7718 |
| | Fifteen feature data (F15) | CB6133 | **0.8631** | **0.8861** | **0.8744** | **0.8517** | **0.8323** |
| | | Cb6133_filtered | 0.8112 | 0.8122 | 0.8117 | 0.8266 | 0.8094 |
| | | CB513 | **0.8334** | 0.7696 | 0.8002 | 0.8212 | **0.8147** |
| | | CAPS10 | **0.7645** | 0.7598 | 0.7621 | **0.7715** | **0.7471** |
| | | CAPS11 | 0.7374 | **0.7819** | 0.7590 | **0.8045** | **0.8021** |

and $maxov(s_1^i, s_2^i)$ is total extent of segments $(s_1^i \text{ and } s_2^i)$, $maxov(s_1^i, s_2^i)$ is calculated using Eq. (3) and $\delta(s_1^i, s_2^i)$ is computed through Eq. (4).

$$maxov(s_1^i, s_2^i) = \left(|s_1^i|, |s_2^i| - minov(s_1^i, s_2^i)\right) \quad (3)$$

$$\delta(s_1^i, s_2^i) = \min\left(maxov(s_1^i, s_2^i) - minov(s_1^i, s_2^i), minov(s_1^i, s_2^i), \left\lfloor \frac{|s_1^i|}{2} \right\rfloor, \left\lfloor \frac{|s_2^i|}{2} \right\rfloor\right) \quad (4)$$

### 5.1.2. Precision

The ratio between the total correctly predicted positive observations to all positive predictions is known as precision. It is also known as Positive Predictive Value (PPV). The precision is defined in Eq. (5)

$$Precision = \frac{T_P}{T_P + F_P} \quad (5)$$

Where, True positive (Tp) show that predicted positive value is actually a positive value and False positive ($F_P$) present that the negative value is incorrectly predicted is positive.

### 5.1.3. Recall

The ratio between the correct positive predictions to all predicted observations of the definite class is known as recall. The computation of recall is shown in Eq. (6).

$$\text{Recall} = \frac{T_P}{T_P + F_N} \quad (6)$$

Where, $F_N$ is a false negative value, means that the positive value is incorrectly predicted is negative.

### 5.1.4. F1 Score

The harmonic mean of precision and recall is known as F1 score. The F1 score is also called Dice Similarity Coefficient (DSC). Its highest possible value is 1, showing ideal recall and precision and if the value of either the recall or the precision is zero then F1 score is zero. The computation of F1 score is shown in Eq. (7).

$$F1 = 2\ X\ \frac{\text{precision} \,.\, \text{recall}}{\text{precision} + \text{recall}} \quad (7)$$

### 5.1.5. Q3 and Q8 accuracies

Several different measurements can be used to gauge the PSSP accuracy, among them Q3, Q8 are the most commonly utilized. Q3 accuracy is defined as the percentage of residue whose secondary structure is predicted to be correct (Ma et al., 2018). The computation of Q3 accuracy is shown in Eq. (8).

$$Q3 = \frac{N_H + N_H + N_C}{N} \times 100 \quad (8)$$

Where, $N_H$ denotes the correctly predicted helix, $N_E$ denotes the correctly predicted strand and $N_C$ denotes the correctly

Table 4
Performance of the Proposed Solution with Decision Tree Classifier on 8 class and 3 class data

| Data | Features | Datasets | Precision | Recall | F1-score | Q8/3 Accuracy | SOV8/3 |
|---|---|---|---|---|---|---|---|
| Eight class data | Five feature data (F5) | CB6133 | 0.6887 | 0.7986 | 0.7396 | 0.8008 | 0.7765 |
| | | Cb6133_filtered | 0.5906 | 0.6899 | 0.6364 | 0.7952 | 0.7635 |
| | | CB513 | **0.7889** | 0.7571 | 0.7727 | 0.7811 | 0.7507 |
| | | CAPS10 | 0.5065 | 0.6005 | 0.5495 | 0.7004 | 0.6708 |
| | | CAPS11 | 0.6888 | **0.7034** | 0.696 | 0.6985 | 0.6798 |
| | Ten feature data (F10) | CB6133 | **0.8097** | 0.8143 | **0.812** | **0.8268** | **0.7888** |
| | | Cb6133_filtered | **0.8101** | 0.8009 | 0.8055 | **0.8056** | **0.7755** |
| | | CB513 | 0.7788 | **0.7951** | **0.7869** | **0.7951** | **0.7604** |
| | | CAPS10 | 0.7069 | **0.7117** | 0.7093 | **0.7411** | **0.7266** |
| | | CAPS11 | **0.7237** | 0.7007 | **0.712** | **0.7445** | **0.7223** |
| | Fifteen feature data (F15) | CB6133 | 0.7957 | **0.8265** | 0.8108 | 0.8242 | 0.7845 |
| | | Cb6133_filtered | 0.7007 | **0.8139** | **0.7531** | 0.8033 | 0.7723 |
| | | CB513 | 0.6876 | 0.7843 | 0.7328 | 0.7901 | 0.7506 |
| | | CAPS10 | **0.7167** | 0.7077 | **0.7122** | 0.7384 | 0.6602 |
| | | CAPS11 | 0.6875 | 0.6978 | 0.6926 | 0.7398 | 0.7187 |
| Three class data | Five feature data (F5) | CB6133 | 0.7777 | 0.7911 | 0.7843 | 0.8663 | 0.8455 |
| | | Cb6133_filtered | 0.7676 | 0.7453 | 0.7563 | 0.8611 | 0.8334 |
| | | CB513 | 0.7986 | 0.7881 | 0.7933 | 0.8704 | 0.8487 |
| | | CAPS10 | 0.7045 | 0.7213 | 0.7128 | 0.7254 | 0.7013 |
| | | CAPS11 | 0.7104 | 0.7321 | 0.7211 | 0.7344 | 0.7154 |
| | Ten feature data (F10) | CB6133 | **0.8315** | 0.8084 | 0.8198 | **0.8903** | **0.8739** |
| | | Cb6133_filtered | **0.8011** | 0.778 | 0.7894 | **0.8896** | **0.8654** |
| | | CB513 | 0.7828 | 0.7597 | 0.7711 | **0.9091** | 0.8166 |
| | | CAPS10 | 0.7619 | 0.7388 | 0.7502 | **0.7819** | 0.7445 |
| | | CAPS11 | 0.7312 | 0.7081 | 0.7195 | **0.7655** | **0.7501** |
| | Fifteen feature data (F15) | CB6133 | 0.8225 | **0.8365** | **0.8294** | 0.8887 | 0.8656 |
| | | Cb6133_filtered | 0.7955 | **0.8334** | **0.8140** | 0.8796 | 0.8507 |
| | | CB513 | **0.8531** | **0.8243** | **0.8384** | 0.8998 | **0.8376** |
| | | CAPS10 | **0.7935** | **0.7771** | **0.7852** | 0.7765 | **0.7555** |
| | | CAPS11 | **0.7786** | **0.7801** | **0.7793** | 0.7575 | 0.7341 |

predicted coil, and $N$ denotes the total number of amino acid residues in a protein. To compute the overall model performance, the average Q3 accuracy is calculated (Eq. (9))

$$Average\ Q3 = \frac{\sum_{i=1}^{n} Q3(X_i)}{n} \quad (9)$$

Where, $n$ denotes the number of protein sequences in test dataset, $X_i$ represents a protein sequence, and Q3($X_i$) represent the Q3 accuracy of $X_i$.

Similarly, in the case of Q8 accuracy computation is performed using Eq. (10).

$$Q8 = \frac{N_E + N_N + N_S + N_T + N_B + N_L + N_G + N_I}{N} \times 100 \quad (10)$$

The $j \in \{N_E + N_H + N_S + N_T + N_B + N_L + N_G + N_I\}$ represent the correctly predicted residues in $j$, the $\{E, H, S, T, B, L, I, G\}$ show the 8 class protein structure. However the $N$ denotes the total number of amino acid residues in a protein. Where, $N_L$ denotes the correctly predicted other residues, $N_B$ denotes the correctly predicted β-bridge, and $N_E$ denotes the correctly predicted β-sheet, $N$ denotes the total number of amino acid residues in a protein.

To calculate the overall model performance, average Q8 accuracy is obtained using Eq. (11)

$$Average\ Q8 = \frac{\sum_{i=1}^{n} Q8(X_i)}{n} \quad (11)$$

Where, $n$ denotes the number of protein sequences in test dataset, $X_i$ represents a protein sequence, and Q8($X_i$) represent the Q8 accuracy of $X_i$.

### 5.2. Competing Methods

The proposed solution is compared with three closely related state-of-the-art methods. These include: Guo et al. (2018), Li et al. (2016), and Kummar et al. (2020).

Guo et al. (2018) use 2DCNN with bidirectional recurrent neural network or bidirectional Long Short Term Memory (BLSTM). This 2C-BRNNs framework contain the four models, 2DCov with bidirectional gated recurrent units (BGRUs) and BLSTM, 2DCNN with BGRUs and BLSTM. They use 2D-CNN for extraction of the local interactions between amino acid residues. For long range interactions between amino acid residues, they use BGRUs or bidirectional LSTM. In their work 2DCov models perform the convolution operation while 2DCNN performs both convolution and pooling tasks. They extract meaningful features form the protein dataset.

Li et al. (2016) use cascaded convolutional neural network and recurrent neural network for the PSSP. Their model consists of four parts, feature embedding layer, multiscale CNN layers with different kernel size, three layers of packed bidirectional gated recurrent unit and at the end two fully connected layers are used. Two types of features are given as the input to the model that are sequence features and profile features. The

Table 5
Performance of the Proposed Solution with MLP Classifier on 8 class and 3 class data

| Data | Features | Datasets | Precision | Recall | F1-score | Q8/3 Accuracy | SOV8/3 |
|---|---|---|---|---|---|---|---|
| Eight class data | Five feature data (F5) | CB6133 | 0.7123 | 0.7111 | 0.7117 | **0.7511** | **0.7418** |
| | | Cb6133_filtered | 0.7066 | **0.7034** | **0.705** | 0.7196 | 0.7038 |
| | | CB513 | 0.5243 | 0.5201 | 0.5222 | 0.5414 | 0.5076 |
| | | CAPS10 | 0.3743 | 0.3739 | 0.3741 | 0.3956 | 0.3702 |
| | | CAPS11 | 0.3908 | 0.3866 | 0.3887 | 0.3963 | 0.3667 |
| | Ten feature data (F10) | CB6133 | 0.6898 | 0.6911 | 0.6904 | 0.7321 | 0.7123 |
| | | Cb6133_filtered | 0.7023 | 0.7011 | 0.7017 | **0.7256** | 0.6978 |
| | | CB513 | 0.5423 | 0.5431 | 0.5427 | 0.5512 | 0.5134 |
| | | CAPS10 | 0.4384 | 0.4298 | 0.4341 | 0.4143 | 0.3914 |
| | | CAPS11 | 0.4076 | 0.3967 | 0.4021 | 0.4159 | 0.4011 |
| | Fifteen feature data (F15) | CB6133 | **0.7345** | **0.7321** | **0.7333** | 0.7455 | 0.7234 |
| | | Cb6133_filtered | **0.7089** | 0.6998 | 0.7043 | 0.7186 | **0.7011** |
| | | CB513 | **0.5524** | **0.5511** | **0.5517** | **0.5666** | **0.5531** |
| | | CAPS10 | **0.4789** | **0.4689** | **0.4738** | **0.4798** | **0.4567** |
| | | CAPS11 | **0.4321** | **0.4412** | **0.4366** | **0.4555** | **0.4453** |
| Three class data | Five feature data (F5) | CB6133 | 0.7231 | **0.7342** | 0.7286 | 0.7555 | 0.7321 |
| | | Cb6133_filtered | 0.7667 | 0.7342 | 0.7501 | 0.7298 | 0.7101 |
| | | CB513 | 0.5432 | 0.5533 | 0.5482 | 0.5531 | 0.5317 |
| | | CAPS10 | **0.5644** | **0.5464** | **0.5553** | 0.4032 | 0.3733 |
| | | CAPS11 | 0.4406 | 0.4021 | 0.4205 | 0.4464 | 0.4313 |
| | Ten feature data (F10) | CB6133 | 0.6987 | 0.6875 | 0.6931 | 0.7543 | 0.7303 |
| | | Cb6133_filtered | 0.7237 | 0.7125 | 0.7181 | **0.7453** | **0.7234** |
| | | CB513 | 0.5654 | 0.5542 | 0.5597 | 0.5687 | 0.5347 |
| | | CAPS10 | 0.4848 | 0.4736 | 0.4791 | 0.4587 | 0.4421 |
| | | CAPS11 | **0.4761** | **0.4649** | **0.4704** | 0.4321 | 0.4123 |
| | Fifteen feature data (F15) | CB6133 | **0.7495** | 0.7291 | **0.7392** | **0.7653** | **0.7523** |
| | | Cb6133_filtered | **0.7878** | **0.7398** | **0.7630** | 0.7234 | 0.7112 |
| | | CB513 | **0.5732** | **0.5791** | **0.5761** | **0.5789** | **0.5436** |
| | | CAPS10 | 0.4899 | 0.4859 | 0.4879 | **0.5023** | **0.4867** |
| | | CAPS11 | 0.4287 | 0.4412 | 0.4349 | **0.4634** | **0.4432** |

function of the embedding layer is to transform the input feature vector into the new vector space. Multiscale CNN with different size of kernels take profile features and new embedded features as the input and extract the multiscale local contextual features. This data is passed to the packed bidirectional gated recurrent unit that gives the global contextual features.

Kummar et al. (2020) utilize CNN and bidirectional RNN for PSSP. Their model consist of four modules: creation of hybrid profile features, the layer for the extraction of local interaction features, the layer for the extraction of long-range interaction features, and the output layer for classification. Position-Specific Scoring Matrices (PSSM) and Hidden Markova Model (HMM) are used for the extraction of discriminating features. In their model 2DCov and 2D max pooling is used for the local interaction between amino acid residues and bidirectional RNN with LSTM and GRUs are used for long range interaction between amino acid residues. In literature, most of the methods use amino acid sequence and position specific-score matrix or profile feature. Using the position-specific scoring matrix to predict the secondary structure of the protein gives better results compared to just utilizing the amino acid sequence. However, it is a computationally expensive task and it also increases the data dimensions (Beckstette et al., 2016). As a result, the computationally power and time increases. The purpose of the proposed method is to find more relevant features and achieve better results by using these features. To evaluate the proposed models, we compare the results with existing methods on four public datasets: CB6133, CB513, CASP10 and CASP11. The comparison is based on the most recent models of PSSP. In this comparison, all the existing methods use standard sequence feature and some other features/information in the form of profile feature/position specific-score matrix. The proposed method uses some (most relevant) standard sequence feature. Still our model achieves better Q8 accuracy.

### 5.3. Experimental attributes and implementation

All experiments are performed on a server with two AMD Opteron 6234 processors, having 128 GB RAM. For the proposed model's implementation, Keras[1] library is used with numpy[2] and sklearn[3] libraries to build and train the proposed models. In the first step, autoencoder is used with one input layer, one hidden layer in the encoder part, one code layer of 17 dimensions, one hidden layer in decoder part and one output layer. Adadelta optimizer is used to train all layers of autoencoder and batch size is set to 64. ReLU activation

[1] https://keras.io
[2] https://numpy.org/
[3] https://scikit-learn.org

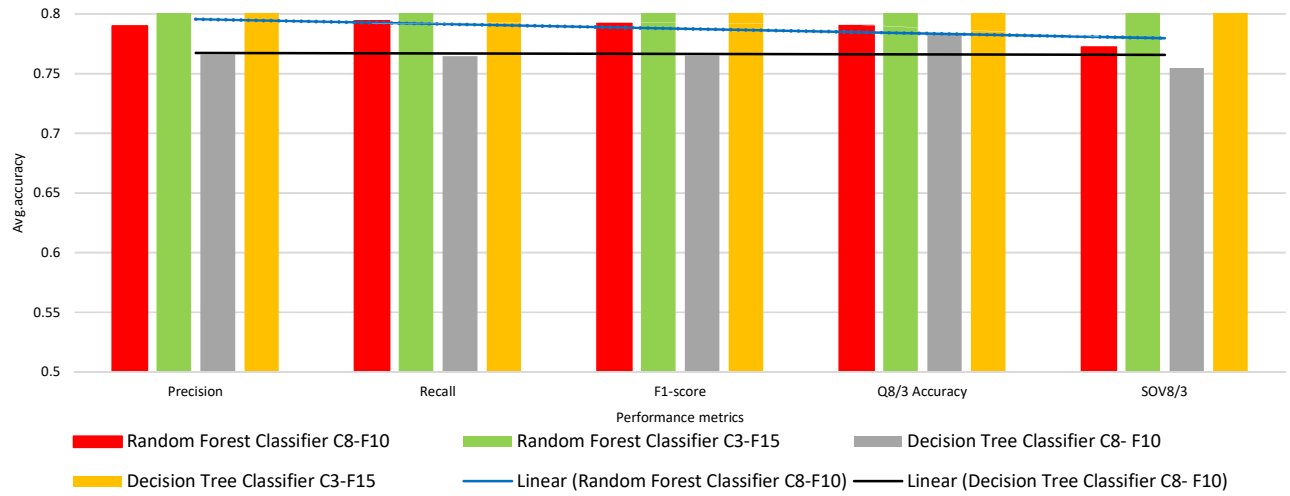



Fig. 3. Comparison of the first and second set of experiments.

function is used in all layers of the encoder and softmax activation function is used in decoder layers.

In the second step, three feature selection techniques, i.e., generic univariate select, REF, and Pearson correlation are used. For generic univariate, select score function lambda and mode percentile are utilized. For REF the step size 1, verbose value 0 estimator and importance getter auto is used.For Pearson's correlation f_regression as a score function is used. Other parameter values are n_samplesint is 100, n_featuresint is 100, n_informativeint 10, n_targetsint 1, biasfloat value 0.0, noisefloat is 0.0, and shufflebool true. In the third step different classifier are utilized. For MLP, three hidden layers of size 100, 75, and 50 are used. Batch size is 64, optimizer is adam, initial learning rate is 0.001, and 200 epochs are used. To avoid over training and over fitting, dropout and early stopping are opted. The hyper parameters of random forest classifier are 100 n_estimators, max_depth is set to none, minimum samples split is 2, and maximum features are set to auto. For decision tree criterion is gini, splitter is best, min_samples_split value is 2, min_samples_leaf value is 1, and min_weight_fraction_leaf value is 0.0.

### 5.4. Evaluation of the Different Selected Features Using Classifiers

The proposed work uses feature extraction and feature selection technique. This experiment has three steps: (1) feature extraction using autoencoder, (2) feature selection using three feature selection techniques and then applying the MI-based aggregation function to take the aggregate of the subset of selected features, and (3) applying classifiers and computing evaluation metrics. As a first step, features are extracted by the autoencoder. As autoencoder is an unsupervised learning, therefore data of 21-dimensional feature map consisting of 21 features of the amino acid residues are passed (without the labels) as the input.

Before data passing to the autoencoder, it is divided into the training set (70%) and testing set (30%) randomly. The sequence amino acid residue is passed as input to the autoencoder. After input layer, the data is passed to the hidden layer and then goes to the code layer. The data here is imbalance, therefore in step three, before applying classifier up-sampling is done after the data split.

#### *5.4.1. TopN best features selection*

In the second step, three feature selection methods, i.e., generic univariate select, REF, and Pearson correlation are applied on the output data of code layer of the autoencoder. From these feature selection methods best (i.e., *TopN*) 15, 10 and 5 features are selected, respectively. Afterwards, the aggregate of subsets features using MI-based aggregation function is computed.

#### *5.4.2. Classification with selected feature subsets*

In step three, the classifiers are applied on the 15 features data, 10 features data, and 5 features data, separately. For each data and classifier 10 experiments are performed, using all the five benchmark datasets using three classifiers, i.e., RF, DT, and MLP. As the classification algorithm of each classifier are different therefore, three classifiers are used in the present work. The performance of the proposed framework is evaluated on five datasets using Q3 accuracy, Q8 accuracy, and SOV score, precision, and recall. The SOV3 denotes the sample overlap of 3 class data, and SOV8 denotes the sample overlap of 8 class data. The performance of the proposed work with RF plugged-in as a classifier is summarized in Table 3. The performance of RF on 5 features and eight class data (C8-F5) shows that the average precision, recall, F1-score, Q8 accuracy, and SOV8 are 77%, 78%, 78%, 75%, and 73%, respectively. However, the average performance of RF on eight class data with 10 features (C8-F10) gives 78% precision, 78% recall, 79% F1-score, 79% Q8 accuracy, and 77% SOV8. Using 15 features and eight class data (C8-F15), the performance of the proposed work on RF classifier is 79% precision, 79% recall, 79% Q8 accuracy, and 76% SOV8. So the experiment shows that using RF on Q8 class data with feature F5, F10, and F15, the proposed work perform better for 15 features data. However, for the three-class data, the performance of the proposed work on 5 features (C3-F5) on

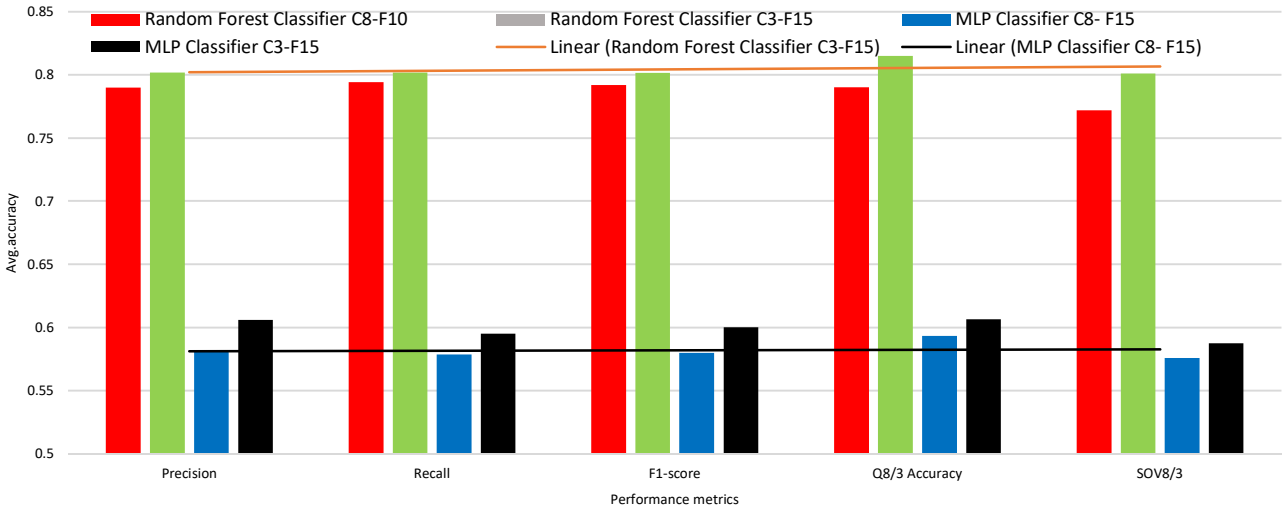



Fig. 4. Comparison of the first and third set of experiments.

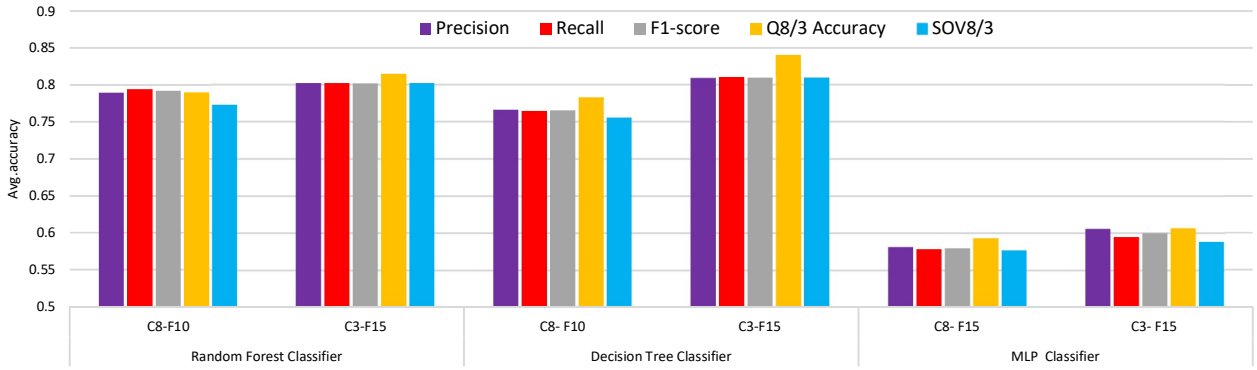
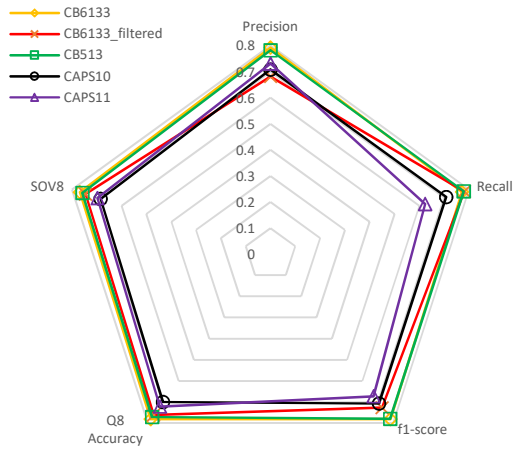
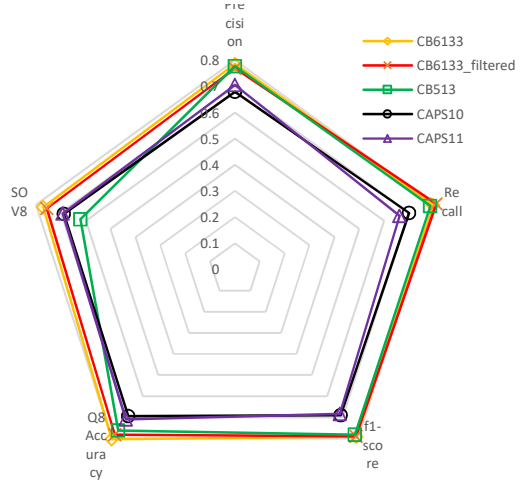



Fig. 5. Classifier performance on Q8-F10, Q3-F10, Q8-15, and Q3-F15.

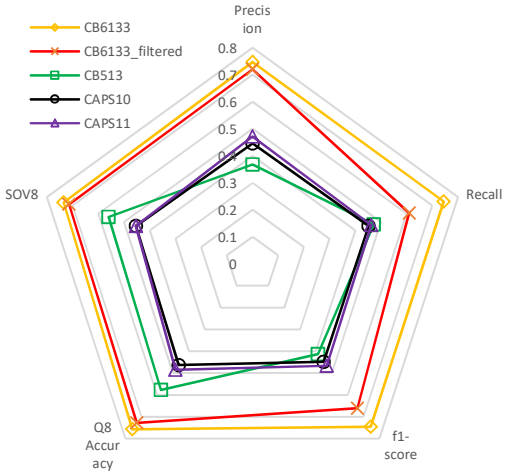
RF classifier gives 76% precision, 79% recall, 77% F1-score, 79% Q3 accuracy, and 78% SOV3. When using 3 class data with 10 features (C3-F10), the performance of the proposed work on RF classifier gives 80% precision, 79% recall, 79% F1-score, 81% Q3 accuracy, and 79% SOV3. The RF classifier gives 80% precision, 80% recall, 80% F1-score, 81% Q3 accuracy, and 80% SOV3 on three-class data with 15 features (C3-F15). Therefore, the first experiment was when using an RF classifier on eight and three-class data with various features set. It's concluded that the proposed work perform better on eight class data with 10 feature set (C8-F10), and three class data with 15 feature set (C3-F15). Table 4 presents the second set of experiments using a decision tree is a classifier on eight and three class data for various feature sets. The performance of the proposed work on eight class data with 5 feature set (C8-F5) gives 65% precision, 70% recall, 67% F1-score, 75% Q3 accuracy, and 72% SOV8. When using the performance of the proposed work on decision tree classifier gives 76% precision, 76% recall, 76% F1-score, and 78% Q8 accuracy, and 75% SOV8. However, for the 15 feature set (C8-F15), the performance of the proposed method on the decision tree gives 71% precision, 76% recall, 74% F1-score, 77% Q8 accuracy, and 73% SOV8. Next, the setting feature set is 5 with three class data (C3-F5). The performance of the proposed work on the decision tree gives 75% precision, 75% recall, 75% F1-score, 81% Q3 accuracy, and 78% SOV3. When using the 10 is the feature set for the three-class data, the classifier gives 78% precision, 75% recall, 77% F1-score, 84% Q3 accuracy, and 81% SOV3. The setting of the feature set is 15 for the three-class data (C3-F15). The proposed work gives 80% precision, 81 recalls, 80% F1-score, 84% Q3 accuracy, and 80% SOV3. The second experiment set shows that the proposed work performs better on the decision tree when using eight class data with 10 features set (C8-F10) and three class data (C3-F3) with 15 features set. The comparisons of the first set of experiments with the second set show that the proposed work performs better on RF and decision tree classifier on (C8-F10) and (C3-F15), respectively, as shown in Fig. 3. Table 5 lists the results of the proposed solution with MLP plugged in as a classifier on eight class and three class data for various feature sets. The average performance of the proposed method on the MLP classifier on the eight class datasets with selected 5 features (C8-F5) is 54% precision, 53% recall, 54%F1-score, 56% Q8 accuracy, and 53% SOV3. When using 10 features for eight class data (C8-F10), the average performance of the proposed method with MLP classifier on five datasets are 55% precision, 55% recall, 55% F1-score, 56% Q8 accuracy, and 54% SOV8. But for C8-F15 the MLP classifier shows 58% precision, 57% recall, 57% F1-score, 59% Q8 accuracy, and 57% SOV8. So by using eight class data with various features set, the




(a)




(b)




(C)

Fig. 6 Performance of the extracted features using different classifiers on 8 class data. (a) RF (b) DT (c) MLP.

proposed work performs better for Q8-F15 on MLP classifier. However, when using three-class data with 5 features set, the MLP gives 60% precision, 59% recall, 60% F1-score, 57% Q3 accuracy, and 55% SOV3. When selected 10 features set for three class data (C3-F10), the average performance of the proposed work is 58% precision, 57 % recall, 58% F1-score, 59% Q3 accuracy, and 56% SOV3. However, for the C3-F15 the MLP classifier gives 60%, 59%, 60%, 60%, and 58% precision, recall, F1-score, Q3 accuracy, and SOV3, respectively on the proposed method. The experiment of the three-class data using 5, 10, and 15 features set presents that the proposed work's performance performs better for the C3-F15. The first and second set of experiments concludes that the proposed work performs better for the random forest classifier. Next, the comparisons is done between the first set of experiment and the third set of experiment is shown in Fig. 4. It is concluded from the comparisons of the experiments that the proposed work performs better on RF classifier for (C8-F10) and (C3-F15). The three types of experiments show that the proposed work performs better on eight class data with 10 features set for the RF classifier. However, it perform better on RF classifier and decision tree classifier for the three class data with 15 feature set (C3-F15) as shown in Fig. 5.

*5.4.3. Evaluation of the Extracted Features Using Classifiers*

In this set of experiments, first, the autoencoder is used for feature extraction, and then the classification is performed for protein secondary structure prediction based on the extracted features. The autoencoder is unsupervised learning model, so the 21 features of the amino acid residues are passed (without the labels) as the input. Results obtained using this experiment on 8 class data are listed in Fig. 6. Decision tree classifier achieves the best recall of 80%, 81%, 79% and 66% on CB6133, CB6133-filtered, CB513, CASP10 and CASP11 and f1-score of 79%, 79%, 78%, 71% and 66% on CB6133, CB6133-filtered, CB513 and CASP11.Decision tree classifier achieves the best Q3 accuracy of 83%, 82% and 81% on CB6133, CB6133-filtered and CB513 dataset. The RF achieves the best Q3 accuracy of 71% and 75% on CASP10 and CASP11 dataset. Decision tree classifier achieves the best Q8 accuracy of 80% and 78% on CB6133 and CB6133-filtered dataset and RF achieved the best Q8 accuracy of 77%, 70% and 72% on CB513, CASP10 and

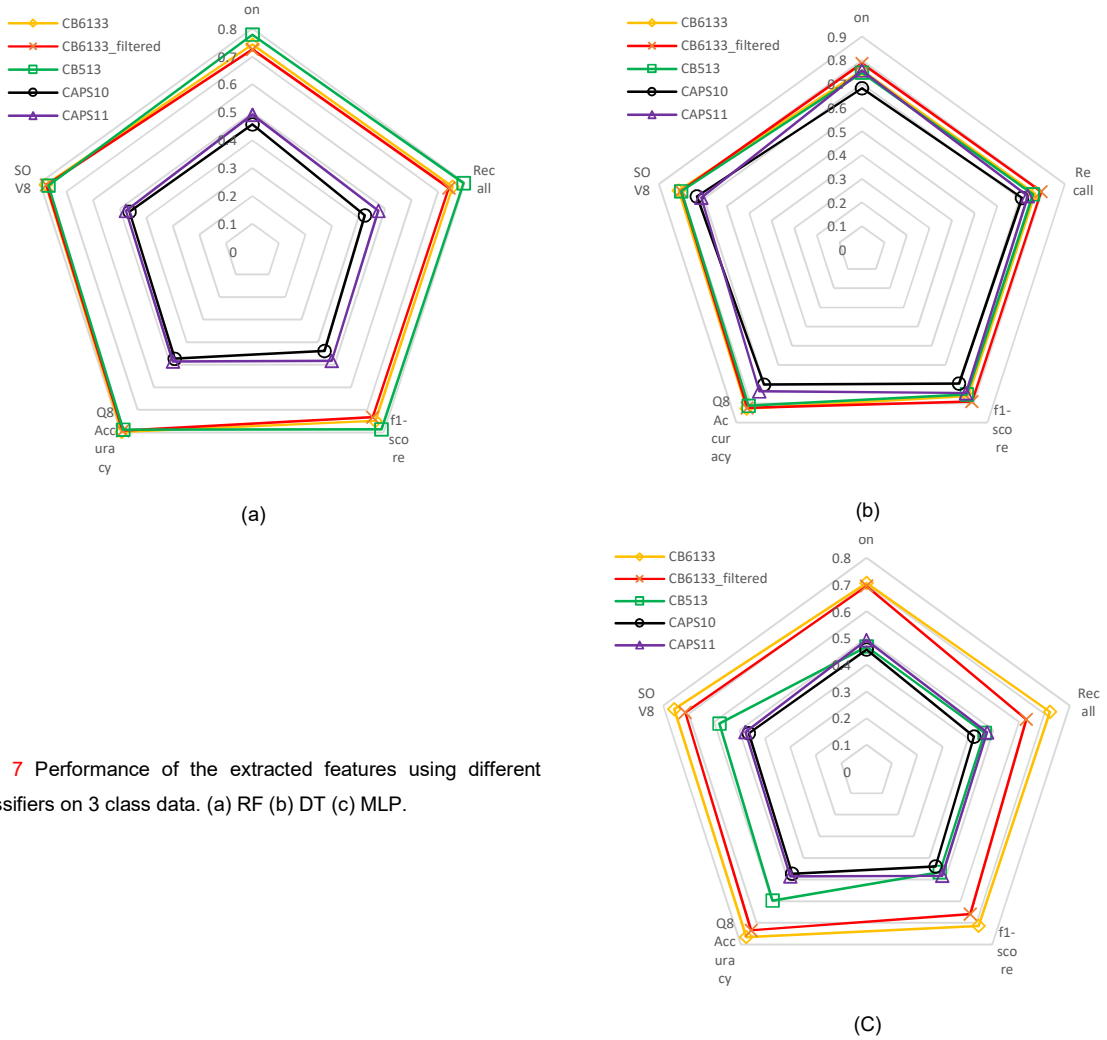



Fig. 7 Performance of the extracted features using different classifiers on 3 class data. (a) RF (b) DT (c) MLP.

CASP11 dataset. The best SOV3 score on CB6133 and CB6133-filtered dataset is 81% and 80%, on CB513 and CASP10 dataset it is 80% and 73%. RF shows the best SOV3 score of 75% on CASP11 dataset. Decision tree classifier give the best SOV8 score of 80% and 78% on CB6133 and CB6133-filtered datasets and RF give the best SOV8 score of 77%, 70%, and 72% on CB513, CASP10 and CASP11 datasets. Two types of experiments are conducted on the proposed work by using five benchmark datasets namely CB6133, CB6133-filtered, CB513, CASP10, and CASP11. These experiments are grouped based on the evaluation of different selected features using multiple classifiers. Both experiments are evaluated by six evaluation metrics.

The first set of experiments used three phased method. These experiments showed that RF give best accuracy of 82%, 76% and 78% on CB513 (10F data), CASP10 (15F data) and CASP11 (15F data). Whereas decision tree classifier achieved better accuracy of 84% on both CB6133 and CB6133-filtered (10F data) datasets. MLP classifier and 5 feature data showed low accuracy on all datasets. The second set of experiments used two stage methods, in these experiments decision tree classifier performed better than other classifiers. The average Q3 accuracy, SOV3, Q8 accuracy and SOV8 on five datasets are shown in Fig. 7. The average Q8 accuracy of Kummar et al. on all datasets is 73%. The proposed framework gives an average Q3 accuracy, average Q8 accuracy, average SOV3 and average SOV8 of 85%, 81%, 78%, and 75%, respectively. Individually, on CB6133, CB6133-filtered, CB513, CASP10 and CASP11 datasets, its Q3 accuracy is 89%, 89%, 91%, 78%, and 77% and its Q8 accuracy is 83%, 81%, 80%, 74%, and 74%. The current method is compared with three state-of-the-art algorithms. Comparing the competing methods based on Q3 accuracy, Q8 accuracy, SOV3 and SOV8 score, on two datasets, i.e., CB6133, CB6133-filter the proposed work has better Q3 accuracy, Q8 accuracy, SOV3 and SOV8 score.

Table 6
Average Results on Five Datasets for the Four Competing Methods

| Methods | Avg. Q3 Accuracy | Avg. SOV3 | Avg. Q8 Accuracy | Avg. SOV8 |
|---|---|---|---|---|
| (Li et al., 2016) | **0.8529** | 0.81 | 0.7388 | 0.7265 |
| (Guo et al., 2018) | 0.7093 | 0.7047 | 0.6779 | 0.6821 |
| (Kumar et al., 2020) | 0.8421 | **0.8136** | 0.7335 | 0.7427 |
| Protein encoder (Proposed) | 0.8472 | 0.8101 | **0.7826** | **0.7547** |

For CB513 the proposed work obtained better Q3 accuracy, Q8 accuracy, and SOV8 score. For CASP11 the proposed work attains better Q3 accuracy and Q8 accuracy. For CASP10 dataset the proposed work preforms lower than the other competing methods. The proposed solution obtains better average Q8 accuracy and SOV8 score among all methods. However, its average Q3 and average SOV3 score is almost similar to the competing methods (see Table 6). On larger datasets, the proposed model performs better, this is because the proposed framework removes redundant and noisy data. Present work used the dimension reduction and feature selection techniques and selects the optimal feature set for the classification. Fig. 8 lists the comparison of the proposed approach with four state-of-the-art methods. The comparisons methods use datasets as follow, "CB6133 dataset is used to train and test the proposed deep learning framework. The CB513, CASP10, CASP11 datasets are only used for testing. CB6133 dataset contains 6133 protein primary sequences and is divided into groups of [0, 5600) training, [5600, 5877) testing and [5877, 6133) validation records for the proposed deep learning framework. Rest of the datasets are entirely used for the testing purpose". In the proposed method, we use autoencoder in first step for feature extraction. When we pass data to autoencoder, the extracted feature of each dataset are different. Furthermore, we use the ensemble feature selection technique in the second step. When we apply feature selection technique on different datasets, they select different features from each dataset. This is the reason underlying not training our model on one dataset and testing on another. To overcome this issue, we split each dataset into test set (30%) and train set (70%) and apply the proposed method on each dataset separately.

**6. Discussion**

The protein plays vital role in our body. It is challenging to analyze the protein dataset, because it is large data containing noise and redundancy. For such a complex data, feature selection play an important role. The present work used a novel approach. The novelty in the proposed work is that it uses unsupervised autoencoder of feature extraction and dimension reduction, the proposed work also used ensemble of three feature selection techniques and used the aggregation

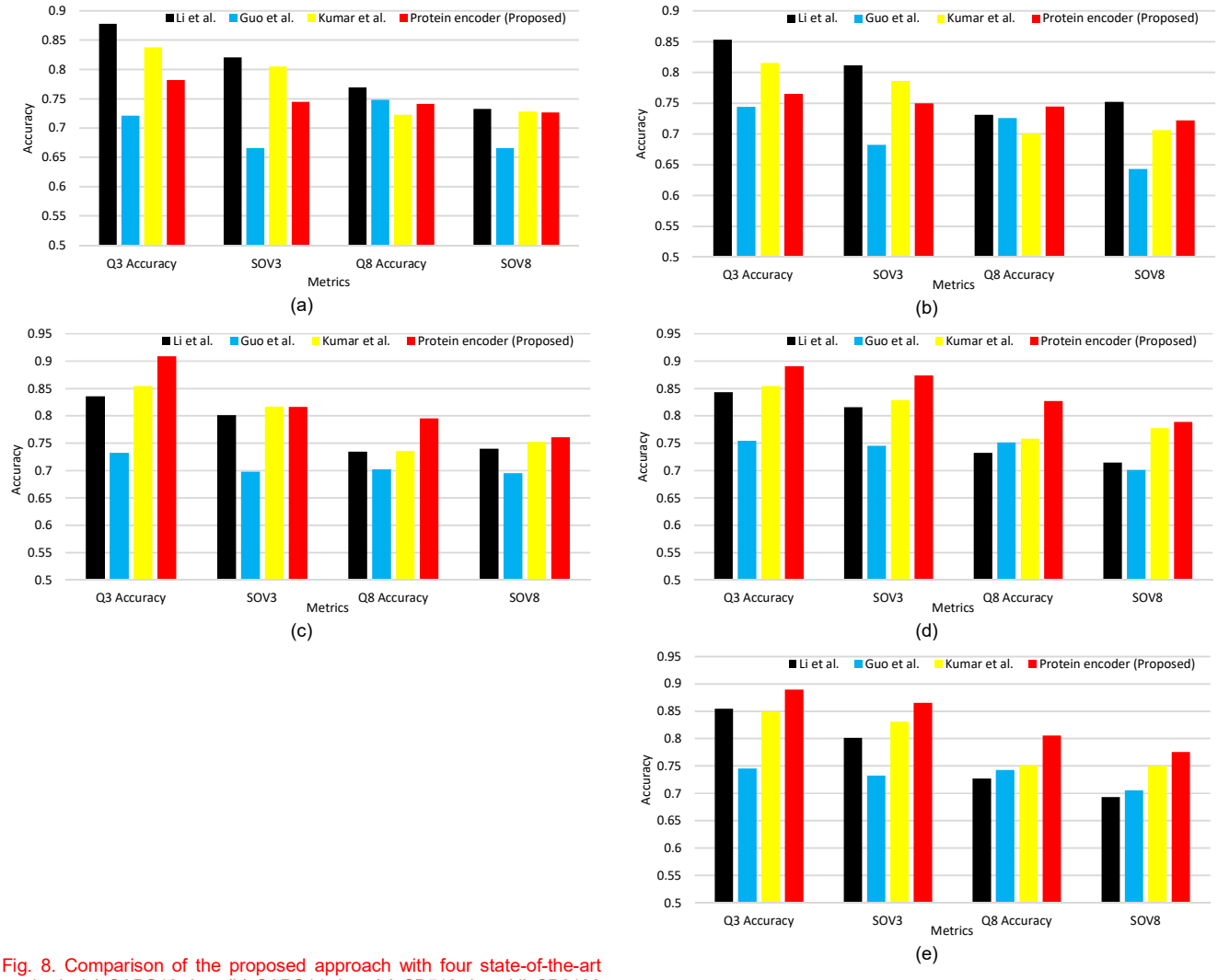



Fig. 8. Comparison of the proposed approach with four state-of-the-art methods (a) CAPS10 data (b) CAPS11 data (c) CB513 data (d) CB6133 data (e) Cb6133_filtered data.

function to combine different features subset. The protein data in the form of matric are passed as an input to the autoencoder.

The reduced extracted features take from the code layer of autoencoder and apply three different feature selection techniques on it. The MI-based aggregation function are used for taking the aggregate of different features subset and find the TopN features. Once the TopN were selected, the three different classifier such as random forest, decision tree and MLP are used for the PSSP. Five benchmark datasets, i.e., CB6133, CB6133-filtered, CB513, CASP10, and CASP11 are used for the evaluation of present work. For Q8 accuracy the RF perform best for the CB6133-filtered, CB513 and CASP11 dataset and decision tree classifier perform best on CB6133 and CASP10 dataset. In second set of experiments decision tree classifier performs better on three class data shown in Fig. 4. The current method is compared with three state-of-the-art algorithms. Comparing the computing method based on the Q3 accuracy, Q8 accuracy, SOV3 and SOV8 score. For two datasets, i.e., CB6133, CB6133-filter the proposed work obtained the better Q3 accuracy, Q8 accuracy, SOV3 and SOV8 score. For CB513 the proposed work obtained the better Q3 accuracy, Q8 accuracy and SOV8 score. For CASP11 the proposed work attains the better Q3 accuracy and Q8 accuracy. For CASP10 dataset the proposed work preforms low then the other comparing methods. The proposed solution obtains the better average Q8 accuracy and SOV8 score among all methods. However, its average Q3 and average SOV3 score is almost same as other competing methods. In large datasets proposed model perform better, the reason is that the proposed framework remove the redundant and noisy data. Present work used the dimension reduction and feature selection techniques and select the optimal feature set for the classification. Among all the comparison method the Li et al. (2016) perform well. It shows overall best Q3 accuracy, SOV3 and SOV8 score on CASP10 and CASP11 dataset and achieve best Q8 accuracy on CASP10 dataset. It performs better in the small datasets. Like the proposed work, Li et al. also used the dimension reduction and feature extraction technique in their model. They use convolution neural network with different kernel size and packed bidirectional gated recurrent unit for the extraction of local and global contextual features.

Other than the novelty and strength of this work, there are a few limitations. The limitation of present work is that it works well on the large dataset because it uses dimension reduction techniques. However, if the data is already in low dimensions then it does not achieve optimum results. Another limitation is that it is a bit challenging to find the optimum feature subset. Hence, a series of experiments are performed here on different features subset to find the optimal feature subset.

## 7. Conclusion

This work presented an efficient model for protein secondary structure prediction from the sequences of amino acid residue. The proposed model used ensemble of three feature selection methods to avoid missing selection of important amino acid residues. For protein secondary structure prediction, the present work used the unsupervised feature extraction technique. For this purpose, autoencoder was used, it is an unsupervised deep learning model. The amino acid residues were given as an input to the autoencoder and feature selection techniques were applied on the output data of the autoencoder code layer. The subset data of different feature selection methods were aggregated by the mutual information-based aggregation function. The proposed model was evaluated on four benchmark datasets. For the evolution of the proposed work four standard and three domain specific evaluation metrics were used. Three classifiers were used for classification. The present work obtained Q8 accuracies of 83%, 81%, 80%, 74% and 84% and Q3 accuracies of 89%, 89%, 91%, 78% and 77% on CB6133, CB6133-filtered, CB513, CASP10, and CASP11 datasets, respectively. The results of the conduced experiment showed that the presented model's accuracy is higher than that of other existing methods and it demonstrates an average increase of 4.3% in Q8 accuracy and 4.7%, 3.5% and 5.5% in Q8 accuracy on CB6133, CB6133-filtered and CB513 datasets. Experiments also demonstrated that the random forest classifier obtained better results on standard evaluation metrics and decision tree classifier showed better results on domain specific evaluation metrics. Moreover, this work was compared with three existing methods and the comparison showed that the proposed framework performed better in majority of the cases. This work can be extended in multiple ways in the future. An extension can be to optimize the feature section through the utility of an evolutionary algorithm. Furthermore, the proposed framework can be used for different other problems, for example, it has a utility in solvent accessibility prediction. Unsupervised feature extraction approach was used in this work, another future direction can be to use supervised feature extraction methods.

**Conflict of interest: None**

**Acknowledgments**

The authors are indebted to the editor and anonymous reviewers for their helpful comments and suggestions. The authors would like to thank GIK Institute for providing research facilities. This work was supported by the GIK Institute graduate program research fund under the GA-1 scheme.

**References**

Araújo, J. D. L., da Cruz, L. B., Ferreira, J. L., da Silva Neto, O. P., Silva, A. C., de Paiva, A. C., & Gattass, M. (2021). An automatic method for segmentation of liver lesions in computed tomography images using deep neural networks. *Expert Systems with Applications*, *180*, 115064.

Aydin, Z., Altunbasak, Y., & Borodovsky, M. (2006). Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. *BMC bioinformatics*, *7*(1), 1-15.

Aydin, Z., Kaynar, O., & Görmez, Y. (2018). Dimensionality reduction for protein secondary structure and solvent

accesibility prediction. *Journal of bioinformatics and computational biology*, *16*(05), 1850020.

Beckstette, M., Homann, R., and Giegerich, R. (2006). Fast index based algorithms and software for matching position specific scoring matrices. BMC Bioinformatics 7, 389. https://doi.org/10.1186/1471-2105-7-389.

Busia, A., & Jaitly, N. (2017). Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction. *arXiv preprint arXiv:1702.03865*.

Burley, S.K., Berman, H.M., Kleywegt, G.J., Markley, J.L., Nakamura, H. and Velankar, S., 2017. Protein Data Bank (PDB): the single global macromolecular structure archive. *Protein Crystallography*, pp.627-641.

Chen, H., Gu, F., & Huang, Z. (2006). Improved Chou-Fasman method for protein secondary structure prediction. *BMC bioinformatics*, *7*(4), 1-11.

Cho, H., & Lee, H. (2019). Biomedical named entity recognition using deep neural networks with contextual information. *BMC bioinformatics*, *20*(1), 1-11.

Dencelin, L. X., & Ramkumar, T. (2016). Analysis of multilayer perceptron machine learning approach in classifying protein secondary structures. *BIOMEDICAL RESEARCH-INDIA*, *27*, S166-S173.

Dowe, D. L., Oliver, J., Dix, T. L., Allison, L., & Wallace, C. S. (1993). A decision graph explanation of protein secondary structure prediction. In IEEE *Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences* Vol. 1, pp. 669-678.

Flynn, T.G., Mercedes, L. and Adolfo, J., 1983. The amino acid sequence of an atrial peptide with potent diuretic and natriuretic properties. *Biochemical and biophysical research communications*, *117*(3), pp.859-865.

Gripon, V., Ortega, A., & Girault, B. (2018). An inside look at deep neural networks using graph signal processing. In IEEE *Information Theory and Applications Workshop (ITA)*, pp. 1-9.

Guo, Y., Li, W., Wang, B., Liu, H., & Zhou, D. (2019). DeepACLSTM: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction. *BMC bioinformatics*, *20*(1), 1-12.

Halim, Z., & Rehan, M. (2020). On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning. *Information Fusion*, *53*, 66-79.

Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems, 5(4), 83-124.

Hoque, N., Singh, M., & Bhattacharyya, D. K. (2018). EFS-MI: an ensemble feature selection method for classification. *Complex & Intelligent Systems*, *4*(2), 105-118.

Hu, X. Z., Long, H. X., Ding, C. J., Gao, S. J., & Hou, R. (2020). Using random forest algorithm to predict super-secondary structure in proteins. *The Journal of Supercomputing*, *76*(5), 3199-3210.

Holley LH, Karplus M. Protein secondary structure prediction with a neural network. Proc Natl Acad Sci U S A. 1989 Jan;86(1):152-6. doi: 10.1073/pnas.86.1.152. PMID: 2911565; PMCID: PMC286422.

Iqbal, S and Halim, Z. "Orienting Conflicted Graph Edges Using Genetic Algorithms to Discover Pathways in Protein-Protein Interaction Networks," IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2021.

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

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, *292*(2), 195-202.

Kabsch, W., & Sander, C. (1983). How good are predictions of protein secondary structure?. *FEBS letters*, *155*(2), 179-182.

Karypis, G. (2006). YASSPP: better kernels and coding schemes lead to improvements in protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, *64*(3), 575-586.

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

Kumar, P., Bankapur, S., & Patil, N. (2020). An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features. *Applied Soft Computing*, *86*, 105926.

Li, Z., & Yu, Y. (2016). Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. *arXiv preprint arXiv:1604.07176*.

Liu, Y., Ma, Y., & Cheng, J. (2017). A novel Group Template Pattern Classifiers (GTPCs) method in protein secondary structure prediction. In IEEE *3rd International Conference on Computer and Communications (ICCC)*, pp. 2713-2717.

Liu, Y., Ma, Y., & Cheng, J. (2017). A novel Group Template Pattern Classifiers (GTPCs) method in protein secondary structure prediction. In IEEE *3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2713-2717.

Liu, Y., Ma, Y., & Cheng, J. (2017). A novel Group Template Pattern Classifiers (GTPCs) method in protein secondary structure prediction. In IEEE *3rd IEEE International Conference on Computer and Communications (ICCC)*, (pp. 2713-2717).

Liu, Z. P., Wu, L. Y., Wang, Y., Zhang, X. S., & Chen, L. (2010). Prediction of protein–RNA binding sites by a random forest method with combined features. *Bioinformatics*, *26*(13), 1616-1622.

Ma, Y., Liu, Y., & Cheng, J. (2018). Protein secondary structure prediction based on data partition and semi-random subspace method. *Scientific reports*, *8*(1), 1-10.

Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE access*, *7*, 19143-19165.

Okun, O., & Priisalu, H. (2007). Random forest for gene expression based cancer classification: overlooked issues. In *Iberian conference on pattern recognition and image analysis*,pp. 483-490, Springer, Berlin, Heidelberg.

Pak, M., & Kim, S. (2017). A review of deep learning in image recognition. In IEEE *4th international conference on computer applications and information processing technology (CAIPT)*, pp. 1-3.

Pollastri, G., Przybylski, D., Rost, B., & Baldi, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, *47*(2), 228-235.

Qian, N., & Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of molecular biology*, *202*(4), 865-884.

Richa, T., Ide, S., Suzuki, R., Ebina, T., & Kuroda, Y. (2017). Fast H-DROP: A thirty times accelerated version of H-DROP for interactive SVM-based prediction of helical domain linkers. *Journal of computer-aided molecular design*, *31*(2), 237-244.

Rost, B., Sander, C., & Schneider, R. (1994). Redefining the goals of protein secondary structure prediction. Journal of molecular biology, 235(1), 13-26.

Selbig, J., Mevissen, T., & Lengauer, T. (1999). Decision tree-based formation of consensus protein secondary structure prediction. *Bioinformatics*, *15*(12), 1039-1046.

Sønderby, S. K., & Winther, O. (2014). Protein secondary structure prediction with long short term memory networks. *arXiv preprint arXiv:1412.7828*.

Song, J., Li, F., Takemoto, K., Haffari, G., Akutsu, T., Chou, K. C., & Webb, G. I. (2018). PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework. *Journal of theoretical biology*, *443*, 125-137.

Torrisi, M., Kaleel, M. and Pollastri, G., 2018. Porter 5: fast, state-of-the-art ab initio prediction of protein secondary structure in 3 and 8 classes. *bioRxiv*, p.289033.

Uzma, F. Al-Obeidat, A. Tubaishat, B. Shah, Z. Halim, "Gene encoder: a feature selection technique through unsupervised deep learning-based clustering for large gene expression data," Neural Computing and Applications, vol. --, 202-.

Uzma, Z. Halim, "Optimizing the DNA fragment assembly using metaheuristic-based overlap layout consensus approach," Applied Soft Computing, Vol. 92, pp. 106256, 2020.

Uzma and Z. Halim, "An ensemble filter-based heuristic approach for cancerous gene expression classification", Knowledge-Based Systems, Vol.234, pp.107560, 2021.

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

Yu, B., Chen, C., Wang, X., Yu, Z., Ma, A., & Liu, B. (2021). Prediction of protein–protein interactions based on elastic net and deep forest. *Expert Systems with Applications*, *176*, 114876.

Zhong, W., Altun, G., Tian, X., Harrison, R., Tai, P. C., & Pan, Y. (2007). Parallel protein secondary structure prediction schemes using Pthread and OpenMP over hyper-threading technology. *The Journal of Supercomputing*, *41*(1), 1-16.