Chapter 3: Protein Secondary Structure Prediction using Sequence Embedding and Bi-LSTM

In this chapter, design and implementation of a framework for representing protein primary sequences as dense embedding vector to predict the secondary structure. This chapter presents improved representation of primary sequences as character embedding and utilizes the bidirectional long short term memory network to retrieve the contextual information for protein secondary structure prediction.

3.1 Introduction

Proteins are essential molecules involved in several functions within the human body, such as cellular signaling, enzymatic activities, antibodies, and many more[10]. The twenty types of amino acids combine to form the primary sequences. Each amino acid has a specific experimental and physicochemical property that decides its structure. Protein structure prediction is essential for medical researchers in drug design and analyzing the effects of mutations on structure and function[11]. Protein is having a hierarchical form of primary structure, secondary structure, tertiary structure, and quaternary structure[12]. Protein primary structure is a linear combination of amino acids. The tertiary structure shows a three dimensional arrangement, which helps in finding their function. The secondary structure connects both primary sequences and tertiary structures[13]. Primary sequences are the basis of secondary structure prediction. The accurate prediction of secondary structure is essential for precise

three-dimensional structure prediction, and beneficial for finding relationships within protein primary sequence and their function [14]. Protein secondary structure mainly has three forms alpha-helix (H), Beta-strand (E), and coil regions (C)[15]. Protein secondary structure prediction from its amino acid sequence is an important problem in molecular biology. Protein structural information is critical for understanding their function, but large number of protein sequences not has their structural information.

The protein primary sequences have some directive which defines their three dimensional arrangement and functionality. We adopt representing primary sequences as dense vectors which consider contextual arrangement of each residue. To find the contextual information, each amino acid residue in primary sequences represented with embedding vectors, which define the three dimensional arrangement and functions of protein [16–18].

Experimental methods for protein structure prediction are X-ray crystallography [19], nuclear magnetic resonance [20], and cryo-electron microscopy[21]. These methods have their advantages and disadvantages. The X-ray diffraction pattern required for xray crystallography. The knowledge of residue conformation and distance between elements that are close to one another used in Nuclear Magnetic Resonance. Prediction of protein structure from scratch using experimental methods not possible because it also requires additional information for molecular structure [10].

Protein secondary structure prediction using experimental techniques is labor-intensive, time-consuming, and costly [22]. Several computational methods used for secondary structure prediction. The statistical techniques based on the propensity of individual residues [23,24]. Support Vector Machine can achieve local minima by

using the local information of residue [25] and Conditional Random Field [26]. These computational methods only depend on local context and ignore non-local context information. Their feature extraction methods are weak because they are handcrafted [27]. In recent years, deep learning models are performing well in several fields, such as speech recognition [28], computer vision[29], and sentiment analysis[30]. Deep neural network methods mainly convolution neural network (CNN) and recurrent neural network (RNN) are also being used in protein secondary structure prediction [6,31,32] because of their automatic representation of raw sequences and learning of nonlinear hidden patterns. Protein sequences have local dependencies [33] as well as long-distance dependencies [6,15,34] between residues for secondary structure prediction. The deep learning-based method applied in protein secondary structure prediction classified as local context-based methods, non-local context-based methods, and a combination of both. Protein secondary structure prediction by utilizing local dependency as well as a long distance dependency used in supervised generative stochastic network method[35]. DCRNN[32] combines the deep convolution and recurrent neural network (DCRNN). CRRNNs [31] used convolution recurrent neural networks (CRRNNs). The CRRNNs combines convolution neural networks with both residual networks and bidirectional recurrent neural networks. CNHH[13] proposed by combining several convolution neural network and highway network. The window-based methods for protein secondary structure prediction capture sequence length 10 – 30. So window-based methods never achieve the state of the art accuracy[10]. The residues far from each other in primary sequences but close to each other in 3-D space. The window based features did not consider long-range

interaction due to the small size of the window[6]. Protein secondary structure prediction accuracy increases by utilizing the increased protein sequences deposited in the protein data bank. A large number of protein sequences and their corresponding secondary structure deposited in protein data bank improved the secondary structure prediction accuracy [36]. Recurrent Neural Networks (RNNs) can learn the sequences. The recurrent neural networks have shortcomings like vanishing and exploding gradients with long sequences and challenging to train with Back-Propagation [37,38]. Recent advancement in the recurrent neural network has resulted in a complex structure. The two improvements in recurrent neural networks are Long Short Term Memory(LSTM)[39], and Gated Recurrent Unit(GRU)[9] which frequently used in prediction and classification in the field of speech and image-related problems. Long short term memory remembers long-range information for time series using a fixed error flow in the network[32]. So the LSTM network performs better than the window-based approach in long sequences [30].

This chapter discuss about: (1) a stacked Bi-LSTM architecture which combines dense embedding vector with bidirectional recurrent neural network for protein secondary structure prediction. The recurrent neural network architecture parameters are evaluated for protein secondary structure prediction. (2) The proposed architecture also compares the performance of Simple Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) for protein secondary structure prediction. (3) The proposed model is evaluated for the three-class secondary structure predictions on three publicly available datasets RS126, CullPDB, and SS.txt. Experiments demonstrate that the combination of character n-gram

embedding vector of primary sequences and stacked Bidirectional Long Short Term Memory networks capture better features to improve the secondary structure prediction.

3.2 Protein Sequence Representation

Deep dense representation of sequences has one of the best performing representation for machine learning methods [40,41]. In this representation, each residue is encoded to stores its interaction information with neighbors. The deep dense representation is motivated by the working structure of human memory, where the information is stored in a "content-addressable". The protein sequences are represented as dense vector, as a deep dense representation for sequences has been efficient way to retrieve contextual information. In this model, each amino acid residue is embedded in a dense n-dimensional vector space. The basic idea behind training such dense vectors is that the amino acid characters are characterized by its contextual information, i.e. neighboring residue.

Given a protein primary sequence with L amino acid residues as $X = x1, x2, x3, \ldots, xL$, where x_i is the m-dimensional dense feature vector of the i^{th} residue, the secondary structure prediction for this protein is formulated as determining $S = s_1, s_2, s_3, \ldots, s_L$ for X where s_i is a Q_3 secondary structure element. In this work, x_i is represented by linear sequence features. Sequence features are utilized to identify the secondary structure for target residue. In order to get deep dense representation of primary sequences, an embedding technique in natural language processing is used. This embedding technique maps amino acid residue as vectors of

real numbers. Specifically, deep dense vector maps amino acid residue from a space with one dimension to a continuous vector space with much lower dimension.

3.3 Dataset Description

The proposed architecture is performing training on a large dataset ss.txt[42]. The training set contains 174,372 protein sequences. The performance of the proposed architecture evaluated using two datasets, CullPDB [43] and RS126[44]. The CullPDB [43] is a non-homologous dataset consist of 6128 protein sequences labeled with secondary structure. The RS126 contains a total of 126 non-homologous globular sequences used as a testing set. The RS126 dataset sequences have an average length of 185, and similarity is below 31%. To ensure the validity of the test result, no two sequences in the training and test sets have a similarity over 30%.

The Define Secondary Structure of Proteins (DSSP) [45] eight types of secondary structure classes, such as 310-helix (G), alpha-helix (H), and pi-helix (I), beta-bridge (B), and beta-strand (E), high curvature loop(S), beta-turn (T) and coil(C). These eight classes converted into three significant types' alpha-helix (H), Beta-strand (E), and coil regions (C) by using standard conversion. All the eight types of the secondary structure of DSSP and their three-class categorization listed in Table 3.1.

Table 3.1: DSSP 8-class to 3-class Conversion

310-helix (G)	H
alpha-helix (H)	H
pi-helix(I)	H
beta-bridge (B)	E
beta-strand (E)	E
loop(S)	C

beta-turn(T)	C
coil(C)	C

3.4 Model Architecture

The proposed architecture is in Figure. 3.1. Firstly, we describe the pre-processing of amino acid sequences, such as integer encoding, padding, and dense embedding representation. Subsequently, a brief description of the Stacked Bi-LSTM layer and the dense layer described.

3.4.1 Feature Extraction

Protein primary sequences have the vocabulary of 20 amino acid characters that is the basis of their encoding. Each amino acid numbered with an integer value in the range of 1-20. The character sequences converted to integer sequences. The protein sequences are of varying length, but the deep learning model accepts the fixed-length. If any sequence exceeds in size, then the remaining character gets discarded. We padded with zeros in the shorter sequence. The string of integers passed to the embedding layer, which changes these integer values to the dense embedding vectors. In the proposed architecture, the Keras embedding layer used that is initialized with random weights and gives an embedding vector for each amino acid.

3.4.2 Deep Network: Bidirectional Long Short Term Memory

A recurrent neural network easily captures long-range interactions between amino acids in protein sequences. The bidirectional recurrent neural network (Bi-RNN) [46] combines information from the past ($h_{backward}$) and future ($h_{forward}$) to predict the current amino acid secondary structure. In this chapter, a similar concept is adopted to

predict the secondary structure of amino acid. The governing equations for Bi-RNN are as follows:

$$h_{forward} = f\left(w_{xh} \cdot h_{t-1} + w_{hh} \cdot x_t\right) \tag{3.1}$$

$$h_{backward} = f\left(w_{xh} \cdot h_{t+1} + w_{hh} \cdot x_t\right) \tag{3.2}$$

$$y = f\left(w_{hy} \times \left[h_{forward}\ ,\ h_{backward}\right]\right) \tag{3.3}$$

Where Wxh, Whh, and Why are weight matrices. xt is current input and ht-1 is previous state.

A deep neural network compressed the dense embedding representation of primary sequences obtained from the embedding layer. The compressed description retains all the information from the character embedding vector. The bidirectional long short term memory(Bi-LSTM) used to capture information from both sides of the central amino acid. In this work, two layers of Bi-LSTM used. First Bi-LSTM, for receiving vectors from the embedding layer and the second Bi-LSTM to generate a new output vector for the one obtained from the first Bi-LSTM.

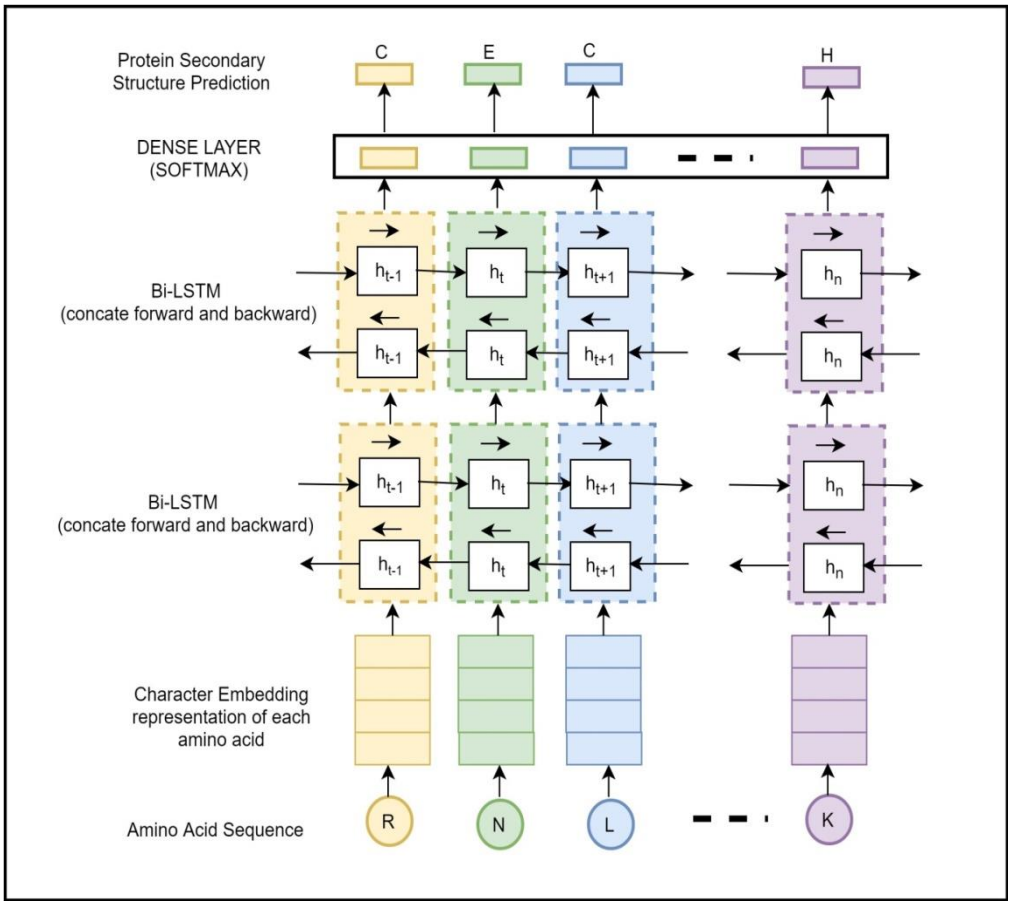

Figure 3.1: Model Architecture of Protein Secondary Structure Prediction

3.4.3 Fully Connected Layer

We have flattened the output vectors obtained after the Bi-LSTM layer to use a dense layer. We pass the deep representation from stacked Bi-LSTM to a fully connected dense layer. To map the input to the final classes of secondary structure, we applied softmax to get a probability score for each of the four alpha-helix, Beta Strand, coil, and no class classes. Softmax having a total sum of probability for each class is one.

$$\sigma(z) = \frac{e^{z_i}}{\sum_{i=1}^{k} e^{z_i}} \tag{3.4}$$

for $i = 1,.....,k$ and $z = z_1.....z_k$

Where z_i is the element of vector z representing the probability of each class, k is equal to four showing the classes alpha-helix, beta-sheet, coil and no-class in vector z.

3.5 Experimental Analysis

Three deep networks Bidirectional Recurrent Neural Network (Bi-RNN), Bidirectional Gated Recurrent Unit (Bi-GRU), and Bidirectional Long Short Term Memory (Bi-LSTM), implemented using Python version 3.6.7. Keras[47] software used at the front end. Keras is an open-source, high-level machine learning API implemented using the Python programming language for CPU as well as GPU. Tensorflow[48] version 1.9 used as back-end due to its excellent numerical computation ability. The dataset is divided into a ratio of 70 to 30, using a statistical sampling method for training and testing. The splitting of data between training and testing is used to ensure the result's accuracy and produce a more generalized model. The recurrent neural network model requires a significant amount of processing time for large datasets due to complex hidden layers with many processing cells. To avoid overfitting, a dropout of 0.1 was adopted in the proposed model. The large mini-batch size results in high computation time for each iteration. If a mini-batch size is selected too small, then the process never converges. The RmsProp optimization [49] is used with a minibatch size of 64 to process the model faster. The categorical cross-entropy used to update the weights and bias. The categorical cross-entropy calculated by the negative loglikelihood loss between the supervised training data and model probability distribution shows a loss between actual and predicted values for the given

training data. In Table 3.2, all the hyperparameters used in the proposed model are listed.

Table 3.2: Configuration of Training Hyper-parameters

Hyper-parameters	**Value**
Cell	64
Dropout	0.1
Recurrent dropout	0.1
Batch size	64
Optimization method	RmsProp
Loss	Categorical_crossentropy
Output activation	Softmax

3.6 Results and Discussion

The proposed architecture implemented the three bidirectional recurrent neural networks such as Bi-RNN, Bi-GRU, and Bi-LSTM for protein secondary structure prediction. Further, three recurrent neural networks performance was evaluated in terms of metrics F-measure, Sensitivity, Specificity, and Accuracy. For training purposes, ss.txt data used, whereas for testing purposes CullPDB and RS126 were used. The results are shown in Table 3.3. It observed that Bi-LSTM is performing better in comparison to others. Bi-LSTM model has higher sensitivity when compared with Bi-RNN and Bi-GRU. The F-measure score is the result of the harmonic mean between precision and recall. The model shows a high F-measure value of more than 90%. Therefore, the precision and recall are equally better for secondary structures

classification. The amount of sensitivity and specificity greater than 90% implies that secondary structure prediction correctly identifies each class. The prediction accuracy is 91.81%.The loss of the Bi-LSTM model plotted with 25 epochs for training and testing datasets. The plot of accuracy for training and testing set for CullPDB and ss.txt is shown in Figures 3.2 and 3.3. The plot of loss shows that the model performance is comparable for both the training set and the testing set from Figures 3.4 and 3.5, The minimum gap between training loss and testing loss shows that the proposed architecture with Bi-LSTM performs well. The values of loss decreases with an increase in training set size.

Table 3.3: Performance of Bi-RNN, Bi-GRU and Bi-LSTM on ss.txt dataset

RNN based Model	**F-measure**	**Sensitivity**	**Specificity**	**Accuracy**
Bi-LSTM	0.9182	0.9134	0.9746	91.81%
Bi-GRU	0.8471	0.8337	0.9595	86.40%
Bi-RNN	0.7840	0.7778	0.9454	80.45%

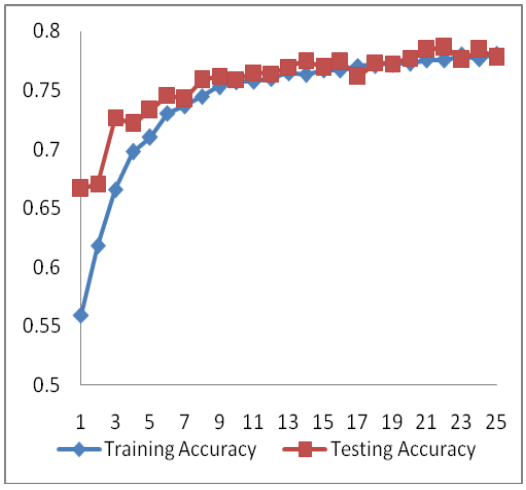

Figure 3.2: Plot of Training Accuracy and testing Accuracy over Epochs for CullPDB

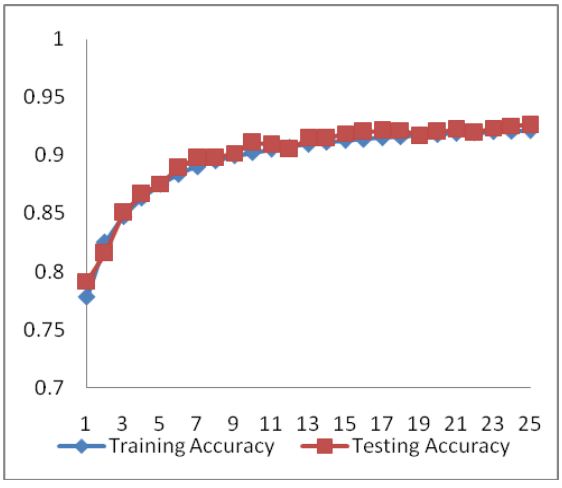

Figure 3.3: Plot of Training Accuracy and testing Accuracy over Epochs for SS.txt

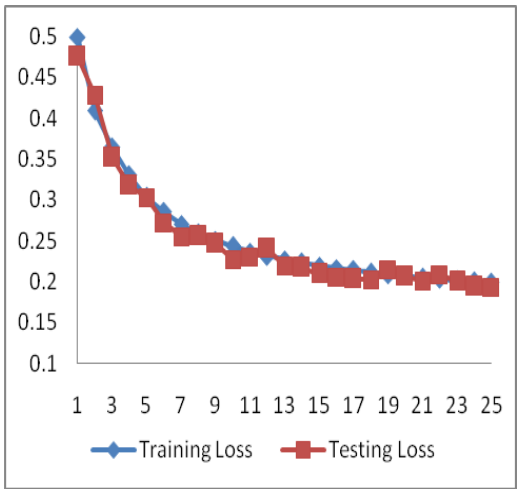

Figure 3.4: plot for Training and Testing Loss over Epochs for SS.txt

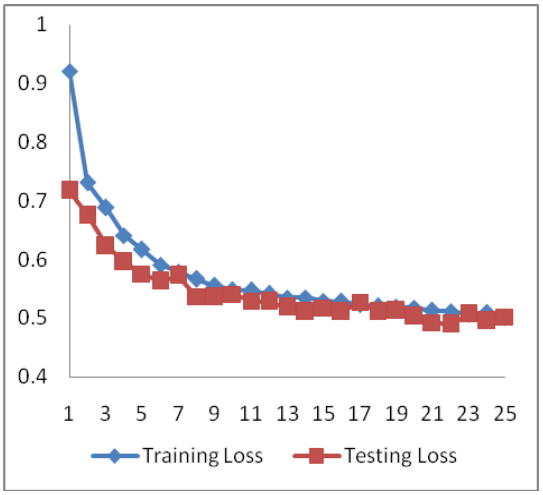

Figure 3.5: plot for Training and Testing Loss over Epochs for CullPDB

The performance of the model was evaluated with Q3 accuracy for three classes of secondary structure. The Q3 accuracy plot with epochs for both dataset CullPDB and ss.txt is shown in figures 3.2 and 3.3, respectively. The value of Q3 for SS.txt is 92.60%, and for CullPDB is 79.16%.

The performance of bidirectional recurrent neural networks (Bi-RNN, Bi-GRU, and Bi-LSTM) and single sequence-based method accuracy are compared in Table 3.4. The Bi-LSTM model shows the improvement in accuracy over Bi-RNN and Bi-GRU. The secondary structure prediction accuracy is significantly higher with the Bi-LSTM model, i.e., 79.16% compared to Bi-GRU, 75.12%, followed by Bi-RNN at 70.93% on CullPDB data. We find that the performance measure of proposed architecture with Bi-LSTM is higher than the other state-of-the-arts methods, i.e., SPIDER3-

Single[4] and PSIpred-Single[50]. SPIDER3-Single used one hot feature vector with Bi-LSTM for secondary structure prediction.

The proposed architecture evaluated for RS126 and compared with the existing state-of-the-art methods such as KB-PROSSP[51], SSpro2.0[36], YASPIN[52], and PHD[53]. The prediction results of the proposed architecture on the RS126 test data sets are reported in Table 3.5. It is very motivating to note that the Q3 accuracy of 80.53% was obtained on the RS126 datasets.

The proposed method can utilize local context and non–local context by using dense embedding vector and Bi-LSTM recurrent neural network. The proposed recurrent architecture's performance evaluated for Bi-LSTM, Bi-GRU, and Bi-RNN to predict the secondary structure. The Bi-LSTM layers have a better result in comparison to Bi-RNN and Bi-GRU during training and testing. We also assess the multilayer Bi-LSTM and find that two-layer Bi-LSTM performs better, but accuracy decreases when using three layers of Bi-LSTM.

Table 3.4: Comparison of the performance of various single sequence based methods on CullPDB dataset

Methods	**Q3(%)**
Bi-LSTM	79.16
Bi-GRU	75.12
Bi-RNN	70.93
SPIDER3-Single	73.24
PSIpred-Single	70.21

Table 3.5: Performance Accuracy comparison with other Methods on Dataset RS126

Methods	**Q3(%)**
KB-PROSSP	77
Sspro2.0	78.13
YASPIN	77.06
PHD	71.4
Our Method	80.53

3.7 Conclusion

In this chapter, a novel architecture is proposed for protein secondary structure prediction. The proposed architecture evaluated for learning capability of Bidirectional Recurrent Neural Network (Bi-RNN), Bidirectional Long Short term Memory (Bi-LSTM), and Bidirectional Gated Recurrent Unit (Bi-GRU). The prediction accuracy of the proposed architecture with Bi-LSTM is better than the existing state-of-the-art single sequence-based methods. We are planning to extend our experiment in a few directions. First, we are planning to utilize n-gram based embedding with stacked bidirectional LSTM. Second, we have the plan to use the contextual capability of bidirectional LSTM and Conditional Random Field. Reinforcement Learning is an unexplored methodology over protein secondary structure prediction, so we are currently conducting experiments by applying Markov Decision Process (MDP).