# Chapter 4: Protein Secondary Structure Prediction using Character n-gram Embedding and Bi-LSTM

This chapter presents an effective method for predicting protein secondary structure using character n-gram embedding and bidirectional long short term memory network. For the prediction of protein secondary structure, primary sequences are represented as dense embedding vectors, which capture the contextual information for each amino acid residues. Further, we examine the most relevant n-gram words in the dataset. Finally, the selected n-gram features are input to embedding layers and construct dense vectors. These dense vectors for n-gram words of amino acid residues combined with bidirectional long short term memory to predict the protein secondary structure from their primary sequences. The proposed method shows a significantly encouraging prediction performance as compared with other state-of-art methods.

## 4.1 Introduction

Proteins are essential biological molecules for performing biological functions in living beings. They play a vital role in the catalysis of biochemical reactions, enzymatic activity, cell communication, and many more [54]. Protein primary sequences are a combination of twenty types of amino acid residues. The primary sequences are varying in length. The amino acid residues far apart from each other in the linear chain some time close in three dimensions. So amino acid residues have a dependency on local as well as long-range contexts for protein secondary structure prediction.

Two amino acid residues combine with peptide bonds between the amino and carboxyl groups. Protein secondary structure prediction is necessary for its function analysis and drug designing. Protein primary structure is a one dimensional amino acid sequence, and tertiary structure defines the structure in three dimensions. Protein functions are highly related to their tertiary structure [55]. Protein secondary structure connects both primary and tertiary structures. The accurate prediction of secondary structure is essential because it results in a more precise three dimensional structure prediction [56]. Protein secondary structure mainly has three forms, alpha-helix (H), Betastrand (E), and coil regions (C).

The recent protein sequencing technique results in a vast number of protein sequences deposited in the protein data bank (PDB). The number of protein sequences is comparably more extensive than predicted proteins secondary structure [10]. Experimental methods have high precision in protein secondary structure prediction. The gap between the number of known sequences and the predicted structure is widening. Thus, there is a greater need than ever before for a reliable computational method to address the problem of protein structure prediction (PSP) directly from the sequence. Since experimental techniques are not efficient, and a large number of protein sequences deposited in the protein data bank, the computational methods required for protein secondary structure prediction [10]. Therefore, accurately predicting protein secondary structure from primary sequences is one of the essential problems in the field of biological computing. Several computational methods utilized for protein secondary structure prediction, such as statistical methods based on the propensity of individual residues[23,57] could not consider contextual information. These methods have Q3 accuracy between 60%-65% approx. Several Machine learning methods used for protein secondary structure prediction, such as Support Vector Machine [25], Conditional Random Field[13], achieved remarkable performances. But machine learning methods depend on local residue statics and ignore the long-range contextual information. Machine learning methods feature extraction methods are weak because they are handcrafted. So feature extraction is essential for a machine learning based predictor. Some feature extraction tools have developed to generate features from primary sequences, such as Pse-in-One [58], BioSeq-Analysis[59], Pse-Analysis[60], and iFeature [61]. Pse-in-One is a webserver using 28 modes to generate feature vectors based on pseudo components. These feature vectors are combined with machine learning methods for analyzing biological processes. The BioSeqAnalysis uses amino acid composition, autocorrelation, pseudo acid composition (PseAAC), profile-based features, and predicted structure features. BioSeq-Analysis2.0. uses Residue composition (One-hot, One-hot (6-bit), Binary (5-bit), Learn from alignments, Position-specific of two residues), physicochemical property, Structure composition, and Evolutionary information. BioSeq-Analysis2.0 incorporates two classification algorithms (Support Vector Machine (SVM), Random Forest (RF), and a sequence labeling algorithm (Conditional Random Fields (CRF). PseAnalysis automatically completes feature extraction, optimizing parameter, model training, cross-validation, and evaluation according to user-provided benchmark dataset for the query sequence. iFeature is a python tool to generate features for protein and peptide sequences. It combines feature clustering, selection, and dimensionality reduction algorithms with machine learning models for analysis and modeling. Deep neural network methods, mainly convolutional and recurrent neural networks, have used in protein secondary structure prediction because of their automatic representation of raw sequences and learning of nonlinear hidden patterns. Protein secondary structures depend on local contexts as well as long-range contexts between amino acid residues. Deep network combined chained conditional neural network with next step conditioning with Q8 accuracy 71.4% [62]. Deep convolution neural network with multiple layer shifts and snitch for protein secondary structure prediction achieves 68.4% Q8 accuracy [63]. Recurrent neural networks combined with profiles perform protein secondary structure prediction with 51.1 % Q8 accuracy [64]. Protein secondary structure prediction using bidirectional long short term memory achieves 67.4% Q8 accuracy [34]. A deep network combines with 2-dimensional CNN, and a two-way recurrent neural network achieves 70.2% Q8 accuracy[27]. The supervised generative stochastic network [35] utilized both local dependency as well as a long-range dependency for protein secondary structure prediction. DCRNN [65]combines cascaded convolution and recurrent neural networks for protein secondary structure prediction. DeepCNF [26] performs secondary structure prediction using conditional random field and shows 82.3% Q3 accuracy and 68.3% Q8 accuracy. The primary sequences are varying in length. The amino acid residues far apart from each other in the linear chain some time close in three dimensions. So amino acid residues have a dependency on local as well as long-range contexts for protein secondary structure prediction.

In recent time the character sequences represented as word embeddings using deep functional architecture. A recursive model used word embeddings which trained over unsupervised morphological analysis [66]. A bidirectional long short term memory (Bi-LSTM) recurrent neural network on characters for embedding arbitrary word types, which showed better performance for language modeling and POS tagging[67]. Character level recurrent neural network methods proposed to represent words for dependency parsing[68], for machine translation, and character to character translation[66,69,70]. The character level recurrent neural network proposed for feature representation and language modeling[71,72]. The convolutional neural network combined with character n-gram have been used for several tasks such as language modeling [73], part-of-speech tagging [74], named entity recognition [69]. The proposed model uses character n-gram embedding for primary sequence representation.

The main contribution of this chapter is: (1) Protein primary sequences are represented as the character n-gram to extract local contexts between amino acid residues. A vector containing counts of character n-grams shows each protein sequence. These character n-grams vectors transform into a low dimensional deep embedding representation. (2) Stacked bidirectional Long Short Term Memory networks used for extracting the non-local context between amino-acid residues. (3) The proposed model is evaluated for the three-class secondary structure predictions on three publicly available datasets ss.txt, RS126, and CASP9. Experiments demonstrate that the combination of character n-gram embedding vector of primary sequences and stacked Bidirectional Long Short Term Memory networks captures better features to improve the secondary structure prediction.

### 4.2 Dataset Description

The proposed architecture used a dataset ss.txt[42]  for training. The dataset has 174,372 protein sequences. Two public datasets, CASP9[75] and RS126[44] , used to evaluate the proposed architecture performance. The CASP9 test set was derived from the 2016 CASP meeting, containing 203 non-redundant proteins. The RS126 is a set of non-homologous 126 globular sequences used as a standard for assessing the model performance. The RS126 data set 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%.

### 4.3  Problem Statement:

We present a simple representation for protein primary sequence as dense embedding vector by utilizing amino acid residues contextual information. A single amino acid residue is represented as character n-gram count vector to show protein primary sequence as dense vector. Protein Secondary Structure prediction from a single sequence of amino acid using character n-gram word embedding vector and Bidirectional Long Short Term Memory (Bi-LSTM) network.

#### 4.3.1  Input features

Protein primary sequence represents the sum of the vector representation of its n-grams. Firstly, generate a vocabulary of n-gram words of size V for amino acid sequences. The aim is to learn a vector representation for each character n-gram word in V. The Keras embedding layer used to represent the sequences as dense embedding vector. The Keras embedding layer initially has some random weights, and during training, learns to embed for each n-gram in the vocabulary list.

### 4.3.2 Outputs:

All the eight types of the secondary structure of Define Secondary Structure of Proteins (DSSP)[45] and their three-class categorization listed in Table 4.1. The DSSP 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.

Table 4.1: 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 |

## 4.4 Methodology

### 4.4.1 Feature Extraction

The primary sequences generated character n-gram words to extract local contextual information. The character n-gram word represents vocabulary set V. Each character n-gram word indexed based on their frequency in the dataset. The n-gram word representation of sequences captures the morphological pattern. Each n-gram words represented as a dense embedding vector. A whole amino acid sequence described with a sum of n-gram embedding vector.

Let $A = \{a_1, a_2, a_3, a_4, a_5, a_{i,}\ldots\ldots\ldots,a_n\}$ be a whole set of amino acid sequences, where n is the number of sequences in dataset, $a_i$ is the $i^{th}$ sequence. $a_i = c_1, c_2,c_{3,}\ldots\ldots\ldots, c_m$, where $m$ represents the length of the $i^{th}$ sequence, $c_m$ indicates the $m^{th}$ character in the $i^{th}$ sequence. Each protein sequence described as the summation of character n-grams word embedding.

### 4.4.2 Deep Network: Bidirectional LSTM

A Recurrent Neural Network easily captures long-range interactions between amino acids in primary sequences. The BRNN[76] combines information from past and future to predict the current amino acid secondary structure. In this chapter, a bidirectional long short term memory (Bi-LSTM) is used to extract in-depth features from primary sequences to predict the secondary structure of a protein.

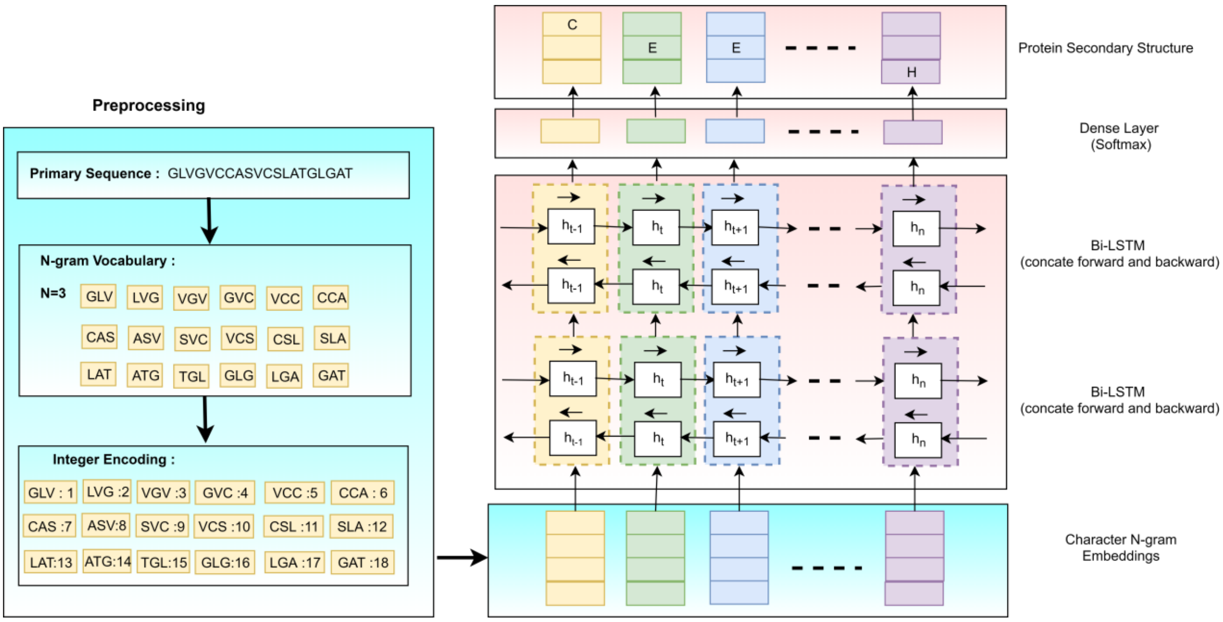
Preprocessing
Primary Sequence : GLVGVCCASVCSLATGLGAT
N-gram Vocabulary :
N=3
GLV LVG VGV GVC VCC CCA
CAS ASV SVC VCS CSL SLA
LAT ATG TGL GLG LGA GAT
Integer Encoding :
GLV : 1 LVG :2 VGV :3 GVC :4 VCC :5 CCA : 6
CAS :7 ASV:8 SVC :9 VCS :10 CSL :11 SLA :12
LAT:13 ATG:14 TGL:15 GLG:16 LGA :17 GAT :18
C
E
E
H
Protein Secondary Structure
Dense Layer
(Softmax)
h_{t-1} h_t h_{t+1} h_n
Bi-LSTM
(concate forward and backward)
Bi-LSTM
(concate forward and backward)
Character N-gram
Embeddings

Figure 4.1: Model Architecture of Protein Secondary Structure Prediction

$$h_{forward} = f\left(w_{xh} \cdot h_{t-1} + w_{hh} \cdot x_t\right) \quad (4.1)$$

$$h_{backward} = f\left(w_{xh} \cdot h_{t+1} + w_{hh} \cdot x_t\right) \quad (4.2)$$

$$y = f\left(w_{hy} \times \left[h_{forward} \;,\; h_{backward}\right]\right) \quad (4.3)$$

Where $W_{xh}$, $W_{hh}$, and $W_{hy}$ are weight matrices. $x_t$ is current input and $h_{t-1}$ is previous state.

After obtaining the deep representation of the primary sequences by using character n-gram embedding as described in the previous section, the proposed architecture uses bidirectional recurrent neural networks (Bi-LSTM). The proposed model uses two layers of Bidirectional LSTM, including one 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 first Bi-LSTM.

### 4.4.3 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 classes of alpha-helix, Beta Strand, coil, and no class. Softmax having a total sum of probability for each class is one.

$$\sigma(\mathrm{z}) = \frac{e^{z_i}}{\sum_{i=1}^{k} e^{z_i}} \tag{4.4}$$

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.

## 4.5 Experimental Details

### 4.5.1 Experimental Setup

Python version 3.6.7 software used. Keras is the 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 [47]. Tensorflow[48] version 1.9 is used as a backend due to its excellent ability in numerical computation.

### 4.5.2 Hyper-parameters

A dropout of 0.1 adopted to avoid overfitting in the proposed model. The large mini-batch size results in high computation time for each iteration. If a mini-batch size is selected to be too small, then the process never converges. The RmsProp[49] optimization is used with a mini-batch size of 64 to process the model at a faster rate. The categorical cross-entropy used to update the weights and bias. The categorical cross-entropy is calculated by the Negative Log-Likelihood loss between the supervised training data and the model probability distribution, which shows loss between actual and predicted values for the given training data. In Table 4.2, all the hyperparameters used in the proposed model listed.

### 4.5.3 Training and Test Strategy

The dataset is divided into a ratio of 70 to 30, using a statistical sampling method for training and testing. This splitting of data between training and testing used to ensure the accuracy of the result and produce a more generalized model. The bidirectional long short term memory network requires a significant amount of processing time for large datasets due to complex hidden layers with a lot of processing cells.

Table 4.2: 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 |

## 4.6 Results and Discussion

The performance of the proposed model evaluated with Q3 accuracy for three classes of secondary structure. The plot of Q3 with epochs for dataset ss.txt shown in figure 4.2. The value of Q3 for SS.txt is 88.45%.

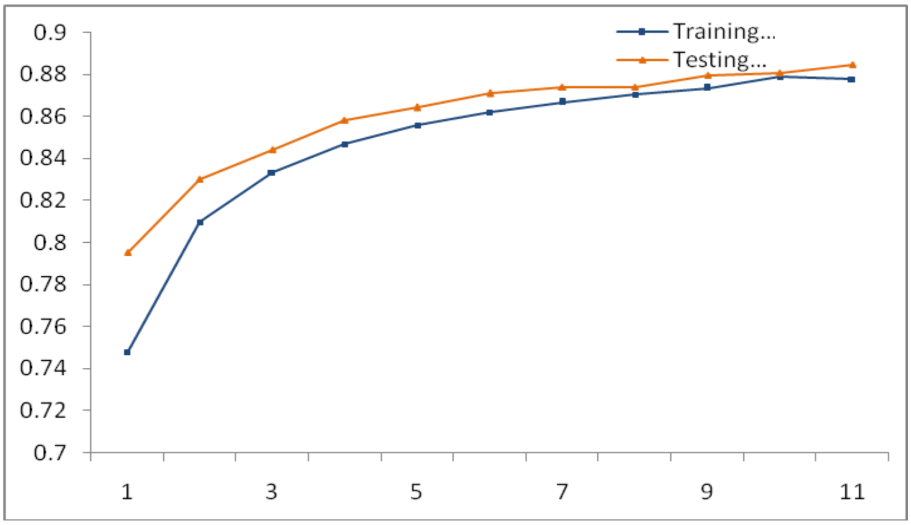

Figure 4.2: Plot for Training and Testing Accuracy over Epochs for ss.txt

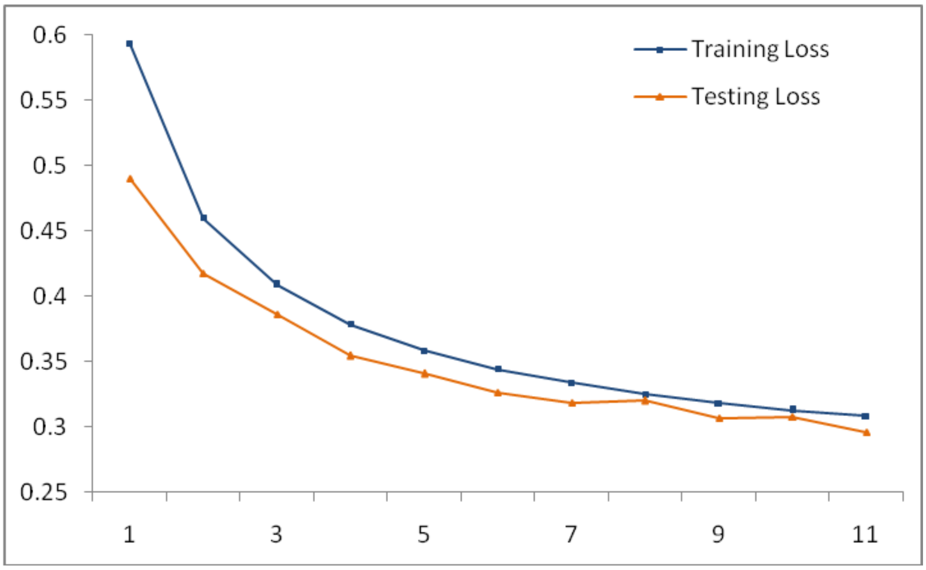

Figure 4.3: Plot for Training and Testing Loss over Epochs for ss.txt

The loss of the proposed model plotted with epochs for training and testing datasets. The plot of loss for training and testing set for ss.txt shown in Figure 4.3. The model performance is comparable for both pieces of the training as well testing set. The minimum gap between training loss and testing loss shows that the proposed architecture with Bi-LSTM is performing well.

The proposed model evaluated for public datasets CASP9 and RS126. The results on both datasets of the model show the effectiveness compared to other state-of-the-art methods. The three single sequence-based secondary structure prediction methods, such as (SPIDER3)[4], JPred4[14], RaptorX[77], MUFOLD-SS[78] and Ensemble LSTM[79] used to compare the model performance on CASP9. These methods based on deep neural networks. As shown in Table 4.3, the Q3 value of our approach is 86.66%, which is higher than SPIDER3, JPred4, MUFOLD-SS, and RaptorX, which were 81.9%, 79.3%, 84.82%, and 81.0% respectively.

Table 4.3: Performance Q3 Accuracy comparison with other methods on CASp9 dataset

| **Methods** | **Q3 (%)** |
|---|---|
| Chargram-Bi-LSTM (Proposed) | 86.66 |
| Ensemble LSTM[79] | 83.3 |
| Spider3[4] | 81.9 |
| Jpred4[80] | 79.3 |
| MUFOLD-SS[78] | 84.82 |
| RaptorX[77] | 81.0 |

Table 4.4: Performance Q3 Accuracy comparison with other methods on RS126 dataset.

| **Methods** | **Q3 (%)** |
|---|---|
| Chargram-Bi-LSTM (Proposed) | 83.48% |
| Psipred[50] | 81.01 |
| PHD[53] | 76.92 |
| Sspro[36] | 77.01 |
| Jpred4[80] | 73.82 |
| MUFOLD-SS[78] | 74.21 |

The proposed architecture evaluated for RS126. The evaluation result compared with the existing state-of-the-art methods such as Psipred[50], PHD[53], Sspro[36], Jpred4[80], and MUFOLD-SS[78]. The prediction results of the proposed architecture on the RS126 test data sets reported in Table 4.4. It is very motivating to note that the Q3 accuracy of 83.48% obtained on the RS126 datasets.

## 4.7 Conclusion

The proposed architecture was evaluated for long sequence learning capability of character n-gram embedding and stacked Bidirectional Long Short term Memory (Bi-LSTM). The experimental analysis shows that the prediction accuracy of the proposed architecture is better than the existing state-of-the-art single sequence-based methods.