Chapter 2: Theoretical Background and Literature Review

This chapter presents the theoretical background related to Protein Secondary Structure Prediction and Deep Learning and a comprehensive literature review related to the thesis's work.

2.1 Protein Secondary Structure Prediction

The protein secondary structure prediction began in 1965. Over more than six decades, the protein secondary structure prediction accuracy improved because the experimental secondary structure determined increased, evolutionary information derived of primary sequences using sequence alignment, improved computational methods such as deep learning algorithms. The deep learning methods' performance improved due to the number of available protein sequences in sequence databases.

Protein secondary structure prediction techniques are classified into three generations[9]. In the first-generation, Protein secondary structures prediction from a primary sequence based on each amino acid residues' statistical information in determining secondary structure element [10,11]. The Chou–Fasman method[11] is the usual first-generation methods, which determine secondary structure using propensities value and heuristic information. The second-generation techniques mainly focus on neighboring residues information and theoretical algorithms such as statistical information [12–14], graph theory [15], neural networks[16,17], logic-based machine learning techniques[18], and nearest neighboring methods [19]. The adjacent

residues information from available protein sequences for secondary structures prediction are data to estimate pairwise, triplet, or longer-segment frequencies. The representative techniques are Garnier Osguthorpe-Robson (GOR) method[20] and the Lim method [21]. In the third generation, the protein secondary structure prediction is determined using evolutionary information such as position-specific scoring matrix derived from homologous sequences[3]. The protein secondary structure is predicted using evolutionary information combined with computational algorithms such as support vector machines[22,23], Bayesian, or hidden semi-Markov network [24,25], and conditional random fields[26]. The artificial neural-network-based models are having a high value of accuracy [5,27,28].

The protein secondary structure prediction performance improves by utilizing better features. The early methods of secondary structure prediction mainly used features from single-residue properties[10,11]. Single residue-based features are followed by the windows based method, which has neighboring residues information [13,14,16], and further evolutionary information derived from multiple homogeneous sequence alignment[3]. The sequence profile information, such as the position-specific scoring matrix, was retrieved using PSI-BLAST [29], derived from homologous sequences. The features from homologous sequences such as sequence profiles improve the secondary structure prediction for three classes above 70% [30,31]. The neural networks achieve better accuracy for large datasets of primary sequences. The profile information such as position-specific scoring matrix and physicochemical properties of a protein used as input to the neural network. Several methods such as HYPROSP [32], PROTEUS[33], DISTILL[34], GOR V [20], SPSSMPred[35], FLOORED [36] ,

and SSpro [37] improve accuracy by using template-based methods. Some methods achieve secondary structure prediction accuracy of more than 90% for proteins by utilizing their homologous sequences' secondary structures. However, many protein sequences are not having secondary structure information of their homologous primary sequences. The SPIDER2[38] method used deep neural networks for secondary structure prediction and gave better accuracy. DeepCNF [39] approach uses deep convolutional neural fields for protein secondary structure prediction. The DeepCNF uses multiple layers of deep convolutional neural networks combined with a conditional random field. The protein secondary structure element prediction depends on a different combination of amino acid residues, such as Helices prediction, which depends on neighbors' hydrogen bonds of primary sequences. The beta-sheet prediction depends on the hydrogen bonds between amino acid residues necessary sequence neighbors. So helices prediction results in better accuracy than sheet residues prediction[9]. The deep learning-based method, such as SPIDER2, shown accuracy for the CASP11 public data set is 86.2% for helices, 75.8% for sheets, and 78.6% for coils [5].

The secondary structure element helices, beta-sheet, and coils have confusion in prediction. The confusion between helices and beta-sheets' prediction is less than the confusion between helices and coil and between beta-sheet and coils [5,40,41]. Several methods are proposed to handle the mess between secondary structure elements [42,43]. The protein secondary structure prediction has short chameleon sequences that behave randomly, which results in different types of secondary structure in other proteins [44, 45]. The chameleon sequences are implicated in

amyloid-type diseases [46]. Several methods [47,48] have proposed that the chameleon protein sequences can accurately predict selecting a sliding window of twenty residues comparably longer than the ten residues window, which is the most prolonged length chameleon sequences [45]. The local interactions between amino acid residues play an essential role in predicting chameleon sequences' secondary structure.

2.2 Deep Neural Network

The multiple hidden layers in a neural network help find the complex nonlinear relationships among attributes and define the complex functions and learning features for classification and prediction problems. The multiple layers enhance the deep neural network's computing capabilities, which results in the development of several methods. In this thesis, recurrent neural networks and their variants, LSTM and GRU, are used to predict the protein secondary structure. The following sections discuss brief descriptions and the working of these deep neural networks.

2.2.1 Recurrent Neural Network

Recurrent neural networks are feed-forward neural networks that process the time series and sequential data with cyclic connections between layers. So, the recurrent neural network utilizes previous data history, which improves the accuracy over a feed-forward neural network.

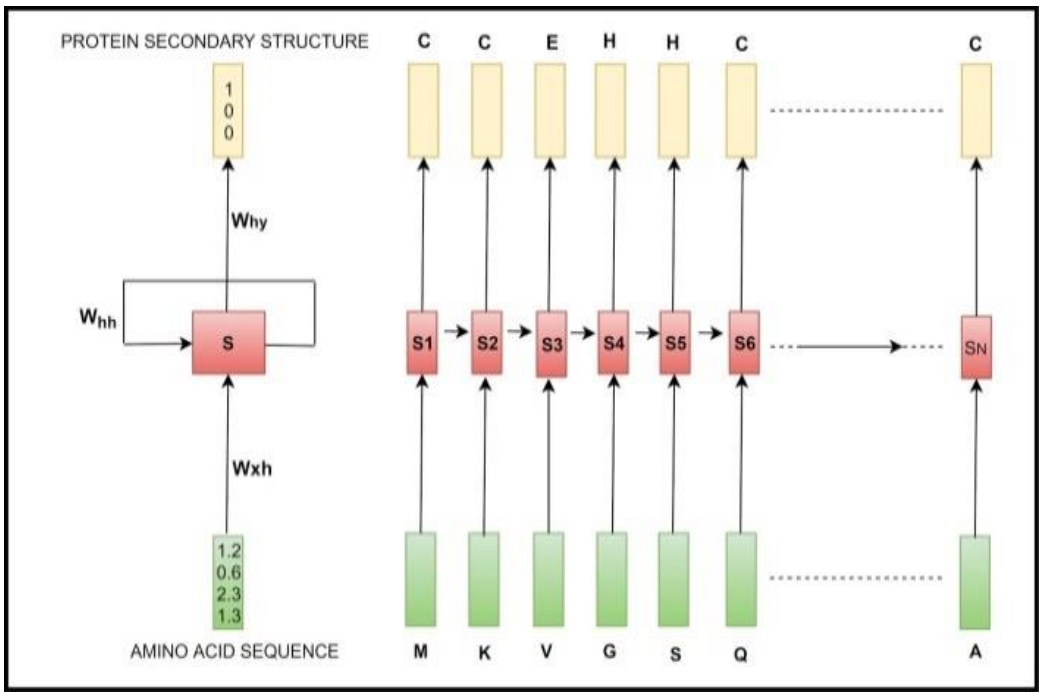

Figure 2.1: Recurrent Neural Network

In Figure 2.1, the recurrent neural network architecture exhibit. Each node in the recurrent neural network shows a time point, the rectangle is a vector, and arrows represent functions. The input vectors are defining amino acid character embedding. The output vectors offer three classes of protein secondary structure. The vectors capture the recurrent neural network state to predict the secondary structure from an amino acid.

The recurrent neural network having an x input vector gives the y output vector. The output vector y is not only predicted by the current input x but also included all history of input feeds to the network in the past. The governing equations for RNN are as follows:

$$h_t = f_h\left(w_{xh}x_t + w_{hh}h_{t-1}\right) \qquad \textbf{(2.1)}$$

$$y_t = f_y\left(h_t \times w_{hy}\right) \qquad \textbf{(2.2)}$$

Where f_h and f_y are activation functions. W_{xh} ,W_{hh} , and W_{hy} are weight matrices. x_t is current input. h_t is current state, y_t is current output and h_{t-1} is previous state.

The recurrent neural network was built to learn the sequences but have the limitations of vanishing and exploding gradients. To overcome the gradient issue in the recurrent neural network, the long short term memory (LSTM) and gated recurrent neural network (GRU) developed.

2.2.2 Gated Recurrent Unit

The Gated recurrent unit (GRU)[9] is a simpler variation of a recurrent neural network that avoids the problem of vanishing gradient by using memory cells in figure 2.2. The update gate (z_t) decides what it should remember from the previous state. The reset gate (r_t) sees the importance of information coming from the former state.

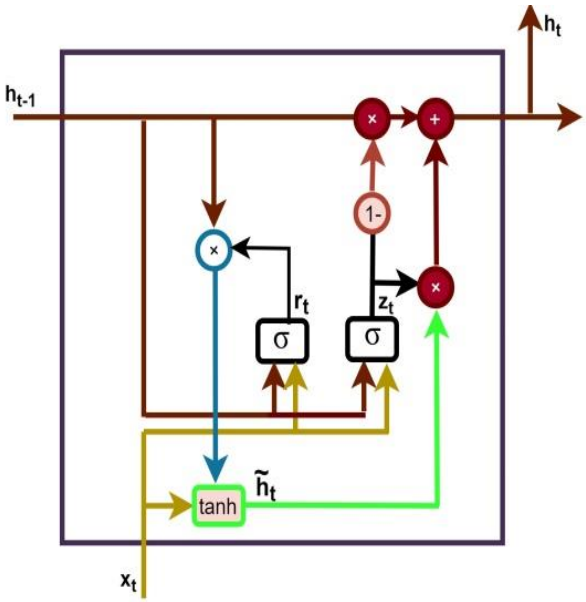

Figure 2.2: Gated Recurrent Unit

The governing equations for the GRU are as follows:

$$z_t = \sigma\left(w_z . x_t + u_z . h_{t-1} + b_z\right) \quad \textbf{(2.3)}$$

$$r_t = \sigma\left(w_r . x_t + u_r . h_{t-1} + b_r\right) \quad \textbf{(2.4)}$$

$$\hat{h}_t = \tanh\left(w_{xh} . x_t + \left(r_t \circ h_{t-1}\right) u_{hh} + b_h\right) \quad \textbf{(2.5)}$$

$$h_t = \left(1 - z_t\right) h_{t-1} + z_t . \hat{h}_t \quad \textbf{(2.6)}$$

Where W_z , W_r and W_{xh} are weight matrices for input vector x_t. U_z , U_r and U_{hh} are weight matrices for previous state h_{t-}. b_z, b_r , and b_h are biases terms of z_t , r_t, and h_t. σ is the sigmoid function.

2.2.3 Long Short Term Memory

Long short term memory (LSTM)[30] uses the operation of selective read, selective forget, and selective write by using the capability of input gate (i_t), forget gate (f_t), and output gate (o_t) for controlling the information. These three gates of LSTM use the current input, previous state, and output selectively by discarding the unnecessary information.

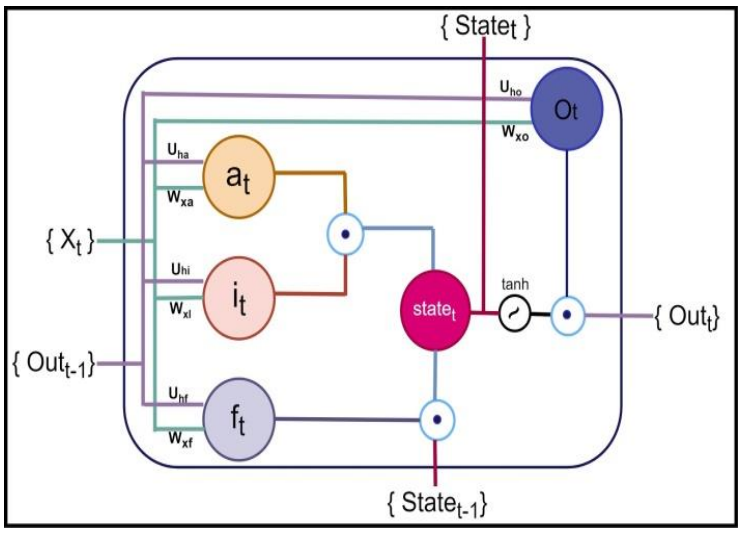

Figure 2.3: Long Short Term Memory Network

The block diagram of LSTM is shown below in Fig. 2.3. The logistic sigmoid activation function and the flow of activation depending on the gates used in hidden layer memory blocks.

The governing equations for LSTM are as follows:

$$a_t = \sigma(w_{xa}.x_t + u_{ha}.Out_{t-1} + b_a) \quad (2.7)$$

$$i_t = \sigma(w_{xi}.x_t + u_{hi}.Out_{t-1} + b_i) \quad (2.8)$$

$$f_t = \sigma(w_{hf}.x_t + u_{hf}.Out_{t-1} + b_f) \quad (2.9)$$

$$o_t = \sigma(w_{ho}.x_t + u_{ho}.Out_{t-1} + b_o) \quad (2.10)$$

$$state = a_t \,\Theta\, i_t + f_t \,\Theta\, state_{t-1} \quad (2.11)$$

$$out = state_t \,\Theta\, o_t \quad (2.12)$$

where W_{xa} , W_{xi} , W_{hf} , W_{ho} are weight matrices for input vector x_t. U_{ha} , U_{hi} , U_{hf}, U_{ho} are weight matrices for previous state Out_{t-}, b_a , b_i , b_f , b_o are the biases terms for each memory gate a_t, i_t, f_t, o_t . Θ signifies the Hadamard product (element-wise product) operation and σ is the sigmoid function.

2.3 Research gaps and Findings:

From the literature review of protein secondary structure prediction following observations are made:

1. **The Irregular behavior of Protein Primary Sequences:**
    - The secondary structure elements have different amino acid residue dependencies, such as helices defined with the hydrogen bond with neighbors, while beta-sheet formed with a hydrogen bond between residues that do not sequence neighbors.
    - The bizarre behavior of chameleon sequences that have a different secondary structure for different primary sequences.
    - The primary sequences are linear but arranged in three-dimensional space, so residues far in the linear chain but close in three-dimensional space.
2. **Complex Primary - Secondary Structure Mapping**
    - The accurate prediction of secondary structure possible with restraints derived from correlated mutations located from Multiple Sequence Alignments of homologous sequences. The multiple sequence alignment needed a large number of homologous sequences**.**
    - The earlier methods used for secondary structure prediction having limitation to learn the long contextual information.
    - Predicting protein secondary structure form primary sequence needed complex dense representation.

- Protein secondary structure prediction depends on local as well as non-local contextual information of primary sequence residues.

2.4 Benchmark Datasets:

RS126: RS126 has 126 protein sequences and comprises 26,846 residues, which is one of the most frequently-used non-homologous dataset developed by Rost and Sandar [33]. The average sequence identity is less than 31% and the average sequence length is 185 residues [61].

CASP9: CASP9 [27] dataset have 203 non-redundant proteins, derived from the 2016 CASP meeting.

CullPDB: The CullPDB (G. Wang & Dunbrack, 2003) is a non homologous dataset consist of 6128 protein sequences labeled with secondary structure.

ss.txt (RCSB PDB): ss.txt is a large dataset provided by RCSB protein data bank. The training set contains 174,372 protein sequences labeled with secondary structure.

2.5 Performance Metrics:

Q score: The estimated performance of PSSP method is often assessed by three-state-per-residue accuracy (Q3) or eight-state-per-residue accuracy (Q8) scoring function which are the simplest and most popular measure methods, and Q score calculates the percent of residues for each secondary structure is correctly predicted.

$$Q_m = 100 \times \frac{1}{N_{res}} \sum_{i=1}^{m} M_{ii} \qquad \textbf{(2.13)}$$

where $m=3$ and $m=8$ is referred as Q_3 and Q_8 accuracy, respectively. N_{res} is the total number of residues, and M_{ii} is correctly predicted number of residues in state i .

The per-state accuracy is the percentage of correctly predicted residues in a particular state, as

$$Q_i = 100 \times \frac{M_{ii}}{obs^i} \qquad \textbf{(2.14)}$$

where obs^i is the number of residues observed in state i.

Precision: Precision is defined as the proportion of instances classified as positive that are really positive.

$$Precision = \frac{TPR}{TPR + FPR} \qquad \textbf{(2.15)}$$

For Secondary Structure Prediction context this precision can also be defined as:

$$Precision = \frac{Number\ of\ Secondary\ Structure\ Element\ Predicted}{Total\ Number\ of\ Secondary\ Structure\ Element\ Predicted} \qquad \textbf{(2.16)}$$

Recall: Recall is defined as the proportion of positive instances that are correctly classified as positive.

$$Recall = \frac{TPR}{TPR + FNR} \qquad \textbf{(2.17)}$$

$$Recall = \frac{Number\ of\ Secondary\ Structure\ Element\ Predicted}{Total\ Number\ of\ Secondary\ Structure\ Element} \qquad \textbf{(2.18)}$$

F-Measure: This measure is approximately the average of the precision and recall when they are close. The two measures are sometimes used together in the F1 Score (or f-measure) to provide a single measurement for a system.

$$F-Measure=\frac{2\times \Pr ecision\times \mathrm{Re} call}{\Pr ecision+\mathrm{Re} call} \qquad \textbf{(2.19)}$$

Specificity: The specificity measure the proportion of negative secondary structure element that are correctly identified and total secondary structure elements.

$$Specificity=\frac{TNR}{TNR+FPR} \qquad \textbf{(2.20)}$$

Sensitivity: The sensitivity measure the proportion of positive secondary structure element that is correctly identified from total secondary structure element.

$$Sensivity=\frac{TPR}{TNR+FPR} \qquad \textbf{(2.21)}$$

2.6 Conclusions

In this chapter, the theoretical backgrounds related to protein secondary structure prediction and deep learning methods as well as literature review are presented. At first, brief overview of protein secondary structure prediction methods, deep learning methods were presented, which provided basis for secondary structure prediction as discussed in subsequent chapters of the thesis. Further, in this chapter a literature survey of prominent approaches for protein secondary structure prediction was discussed and research gaps were identified.