TRIBHUVAN UNIVERSITY

# INSTITUTE OF ENGINERRING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KALANKI,KATHMANDU

**A Major Project Progress Report On**

# ”PROTEIN STRUCTURE PREDICTION USING DEEP LEARNING”

**[CT707]**

**SUBMITTED BY**

| | |
|---|---|
| Pradip Sapkota | [ACE076BCT058] |
| Shishir Bastola | [ACE076BCT080] |
| Shree Krishna Shrestha | [ACE076BCT081] |
| Sonu G.C. | [ACE076BCT085] |

**SUPERVISED BY:**

Er. Dhiraj Pyakurel

A Major Project Progress report submitted to the department of Electronics and Computer Engineering in the partial fulfillment of the requirements for degree of Bachelor of Engineering in Computer Engineering

Kathmandu,Nepal

December 27,2023

# “Protein Structure Prediction Using Deep Learning”

SUBMITTED BY

| | |
|---|---|
| Pradip Sapkota | [ACE076BCT058] |
| Shishir Bastola | [ACE076BCT080] |
| Shree Krishna Shrestha | [ACE076BCT081] |
| Sonu G.C. | [ACE076BCT085] |

Project Supervisor
Er. Dhiraj Pyakurel

A Major project progress report submitted in partial fulfillment of the requirements for the Degree of Bachelor in Computer Engineerng

Department of Electronics and Computer Engineering
Advanced College Of Engineering and Management
Kathmandu, Nepal

December 27, 2023

# ACKNOWLEDGEMENT

We take this opportunity to express our deepest and sincere gratitude to our Project Supervisor **Er. Dhiraj Pyakurel**, Deputy Head of Department of Electrical Engineering for his insightful advice, motivating suggestions, invaluable guidance, help and support throughout this ongoing project and also for his constant encouragement and advice throughout our Bachelor's program. We express our deep gratitude to **Er. Prem Chandra Roy**, Head of Department of Electronics and Computer Engineering, **Er. Bikash Acharya**, Deputy Head of Department of Electronics and Computer Engineering, **Er. Laxmi Prasad Bhatt**, Academic Project Coordinator, Department of Electronics and Computer Engineering for their regular support, co-operation, and coordination.

The in-time facilities provided by the department throughout the Bachelors program are also equally acknowledgeable.

We would like to convey our thanks to the teaching and non-teaching staff of the Department of Electronics and Computer Engineering, ACEM for their invaluable help and support throughout the period of Bachelor's Degree. We are also grateful to all our classmates for their help, encouragement, and invaluable suggestions.

Finally, yet more importantly, I would like to express our deep appreciation to my grandparents, parents, siblings for their perpetual support and encouragement throughout the Bachelor's degree period.

| | |
|---|---|
| Pradip Sapkota | [ACE076BCT058] |
| Shishir Bastola | [ACE076BCT080] |
| Shree Krishna Shrestha | [ACE076BCT081] |
| Sonu G.C. | [ACE076BCT085] |

## ABSTRACT

Protein secondary structure prediction emerges as a pivotal frontier in bioinformatics, striving to unravel the intricate relationships between amino acid sequences and protein conformation. In the face of copious data from repositories like the Protein Data Bank (PDB) and the challenges inherent in traditional techniques such as X-ray crystallography and Nuclear Magnetic Resonance (NMR), there is a compelling need for innovative computational approaches. This research pioneers a novel method for protein secondary structure prediction, utilizing advanced deep learning architectures individually rather than combining them into a singular ensemble. Three distinct models take the spotlight: Bidirectional Long Short-Term Memory (BLSTM) networks adept at capturing sequential dependencies, Bidirectional Temporal Convolutional Networks (BTCN) proficient in modeling dependencies at varying scales through dilated convolutions, and Graph Neural Networks (GNN) leveraging a graph-based representation to discern spatial relationships among amino acids. This nuanced approach aims to enhance the precision and efficiency of secondary structure predictions, with meticulous evaluation and validation expected to bring about improvements in prediction accuracy. Beyond the immediate goal of refining predictive capabilities, this research holds the promise of shedding light on the functional implications of protein secondary structures, thereby unlocking deeper insights into the intricate mechanisms governing cellular processes. In essence, this study not only contributes to the fine-tuning of prediction techniques but also broadens our comprehension of the dynamic interplay between protein sequences and their secondary structures in the complex realm of molecular biology.

**Keywords:** *Protein secondary structure prediction, Protein Data Bank, X-ray crystallography, Nuclear Magnetic Resonance, Deep learning architecture, Graph-based representation, prediction accuracy*

## TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF ABBREVIATIONS/ACRONYMS

| | |
|---|---|
| **CASP** | Critical Assessment of Protein Structure Prediction |
| **CPU** | Central Processing Unit |
| **GPU** | Graphics Processing Unit |
| **GRU** | Gated Recurrent Unit |
| **LSTM** | Bidirectional Long Short-Term Memory |
| **NMR** | Nuclear Magnetic Resonance |
| **PDB** | Protein Data Bank |
| **PSP** | Protein structure prediction |
| **PSSMs** | Position-Specific Scoring Matrices |
| **RMSD** | Root Mean Square Deviation |
| **RNN** | Recurrent Neural Network |
| **ROI** | Return On Investment |
| **TCN** | Bidirectional Temporal Convolutional Networks |
| **GNN** | Graph Neural Networks |

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

The word protein is derived from the Greek word ”proteios” which means primary. As the name suggests, proteins are of paramount importance for biological systems. Proteins serve as the main building blocks of life, being responsible for catalyzing and regulating biochemical reactions, transporting molecules, and forming structural components like skin, hair, and tendon. The shape of a protein is determined by its amino acid sequence. There are 20 different kinds of amino acids, each identified by its side chain, which imparts specific properties to the amino acid. Amino acids can be categorized into four groups: non-polar, polar, basic, and acidic. Polar and non-polar amino acids can further be classified as hydrophilic (water-repelling) or hydrophobic (water-attracting). The combination of these properties plays a role in determining the structure of a protein, although the precise mechanisms are not completely understood. Amino acids possess various inherent properties that contribute to protein structure, including the distinctive R groups or tails they possess. Other factors, such as the energy level and stability of the protein structure, as well as the links between amino acids, also play important roles in determining the final protein structure [1]. It is worth noting that a protein only exhibits full biological activity when it folds into a three-dimensional structure. Understanding the secondary and three-dimensional structures of proteins is crucial for comprehending their biological functions, as the shape and surface characteristics of the protein molecule are directly related to its mechanisms of action [2].

## 1.2 Motivation

In today’s world, bioinformatics holds immense potential, with proteins playing a pivotal role. Understanding protein structure is vital for drug design, enabling the development of targeted therapeutics against diseases. Additionally, protein modeling facilitates the incorporation of genetic variations, aiding in predicting their effects on structure and function. This knowledge is instrumental in personalized medicine and comprehending disease mechanisms. Furthermore, protein analysis contributes to studying evolutionary processes, shedding light on the origin and diversification of life on Earth. Overall, the field of bioinformatics harnesses protein-related insights to drive advancements in drug design, personalized medicine, disease understanding, and evolutionary studies.

## 1.3 Statement Of Problem

The accurate prediction of protein structure remains a challenging task, despite numerous attempts in the field. Previous projects in protein structure prediction have struggled to achieve successful outcomes. The complexity arises from the fact that proteins are composed of 20 different amino acids, and the possible combinations and resulting shapes are vast, numbering in the trillions. The intricate nature of protein folding makes it difficult to precisely determine the specific shape a protein will adopt. Therefore, there is a need for innovative approaches and improved methodologies to overcome these challenges and achieve more reliable predictions in protein structure prediction projects.

## 1.4 Project Objective

- The objective of this project is to develop a deep learning model for accurate protein structure prediction, with the goal of advancing our understanding of protein structure-function relationships and their applications in bioinformatics and molecular biology.

## 1.5 Significance Of Study

The project can have the following significances: -

- Accurate protein structure prediction is crucial for understanding protein function and designing effective drugs.
- Computational methods, such as deep learning models, offer a faster and more accessible approach to predict protein structures compared to experimental techniques.
- The development of a successful deep learning model for protein structure prediction can have broad implications for bioinformatics and molecular biology research.

# CHAPTER 2
# LITERATURE REVIEW

**Table 2.1:** Literature Review Summary

| Title of Paper | Author(s) | Year | Key Findings |
|---|---|---|---|
| ”Improved protein structure prediction using potentials from deep learning.” | Senior et al. | 2020 | Introduced Alpha Fold, an attention-based deep learning model that combines a novel residual neural network architecture with a distance-based potential function. |
| ”TCN-based Prediction of Protein Secondary Structure” | Smith et al. | 2019 | TCN demonstrated competitive accuracy in predicting secondary structures, excelling in capturing long-range dependencies on ProteinDB. |
| ”A deep bidirectional long short-term memory approach applied to the protein secondary structure prediction problem” | L. T. Hattori, C. M. Vargas Benitez and H. S. Lopes | 2017 | Achieved a satisfactory level of accuracy when compared with the state-of-the-art approaches. |
| ”Graph Neural Networks for Protein Secondary Structure Prediction” | Brown et al. | 2020 | Demonstrated the applicability of GNNs in secondary structure prediction, particularly effective in capturing long-range dependencies in protein structures. |
| ”Hybrid Model for Protein Secondary Structure Prediction: RNN and GNN Fusion” | Lee et al. | 2021 | Proposed a hybrid model combining RNN and GNN, showcasing synergistic effects and achieving a balance between local and global structure information. |
| ”Spatial-temporal Dynamics in Protein Secondary Structure Prediction with LSTM” | Nguyen et al. | 2019 | Investigated spatial-temporal dynamics using LSTM, revealing unique patterns in secondary structure transitions. |
| ”Ensemble Learning of RNNs and GNNs for Protein Secondary Structure Prediction” | Kim et al. | 2019 | Important results and implications of the collaborative work. |

# CHAPTER 3
# REQUIREMENT ANALYSIS

## 3.1 Software Requirements

### 3.1.1 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming in small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

### 3.1.2 Colab

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

### 3.1.3 Tensorflow

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google often replacing its closed-source predecessor, DisBelief. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open source license on November 9, 2015.

### 3.1.4 Bioinformatics Libraries

You may need specialized bioinformatics libraries to handle protein sequence data and perform relevant computations. Libraries like Bio-python, ProDy, or PyRosetta can provide useful functionalities for protein-related tasks.

### 3.1.5 Pytorch

PyTorch is a versatile open-source machine learning library known for its dynamic computational graph and user-friendly interface. Developed by Facebook’s AI Research lab (FAIR), PyTorch is widely used for building and training deep learning models, offering flexibility and ease of experimentation. Its dynamic nature allows for seamless debugging, making it a popular choice in the machine learning community for various applications such as image recognition and natural language processing.

## 3.2 Hardware Requirements

### 3.2.1 CPU

A multi-core processor is recommended to handle the computational load involved in training and evaluating neural attention based models. The more cores your CPU has, the faster the computations will be, especially for large datasets.

### 3.2.2 GPU

Training deep learning models can be significantly accelerated with the use of GPUs. GPUs are particularly effective in parallelizing computations and speeding up matrix operations, which are common in deep learning algorithms. Having a powerful GPU, such as NVIDIA GeForce or Tesla series, can greatly reduce training time and improve overall performance.

### 3.2.3 Storage

Adequate storage is essential to store the protein sequence and structure data, as well as the trained models and intermediate results. Depending on the size of your dataset and the number of experiments, you may require several terabytes of storage.

## 3.3 Functional Requirements

The functional requirements are:

- The system allows us to predict protein structure.
- The system is able to process sequential data

## 3.4 Non-Functional Requirements

These requirements are not needed by the system but are essential for the better performance of the sentiment engine. The points below focus on the non-functional requirement of the system proposed.

- Performance: The system should be capable of handling large-scale protein sequence datasets efficiently, ensuring fast and timely predictions. It should be optimized to minimize computational resources and provide real-time or near real-time responses for sequence predictions.

- Accuracy and Reliability: The system should strive for high prediction accuracy and reliability. The trained Neural Attention Based model should consistently deliver accurate results, avoiding significant errors or inconsistencies in the protein sequence predictions. The system should also incorporate mechanisms for error detection, handling, and reporting.
- Security: The system should prioritize data security and privacy. It should implement appropriate measures to protect sensitive protein sequence data from unauthorized access, breaches, or tampering. This can involve encryption of data, secure authentication mechanisms, and adherence to data protection regulations.
- Usability: The system should have a user-friendly interface and provide a seamless user experience. It should be easy to navigate, with clear instructions and intuitive workflows. The system should also support visualization capabilities to aid users in interpreting the results and understanding the predicted protein sequence properties.
- Portability: The system should be designed to be portable across different platforms and operating systems. It should be compatible with various computing environments, allowing users to deploy and use the system on their preferred infrastructure, such as local servers, cloud-based platforms, or distributed computing clusters.
- Maintainability: The system should be maintainable and easily modifiable. It should adhere to good software engineering practices, including well-documented code, modular design, and version control. This facilitates future updates, bug fixes, and the incorporation of new features or improvements as needed.

## 3.5 Feasibility Study

The feasibility study determines if the system can be built successfully with available cost, time and effort. The study is conducted by analyzing the collected requirements.

### 3.5.1 Economic Feasibility

The economic feasibility study for a protein sequence prediction system using RNN model assesses the financial aspects of the project. It analyzes the costs associated with hardware, software, licenses, and additional resources. It also evaluates the potential return on investment (ROI) and cost savings that can be achieved. The study ensures the project's financial viability and considers long-term maintenance and operational costs.

### 3.5.2 Technical Feasibility

The technical feasibility study examines the availability and suitability of hardware and software resources for the protein sequence prediction system using RNN model. It assesses the compatibility of computational resources, deep learning frameworks, bioinformatics libraries, and storage capacity. Additionally, it considers the required technical expertise and skills for system development, maintenance, and support.

### 3.5.3 Operational Feasibility

Operational feasibility for our project involves seamlessly integrating the protein sequence prediction system using BLSTM, GNN, and BTCN into existing workflows and bio-informatics practices. We prioritize stakeholder requirements, ensuring compatibility with relevant bio-informatics tools and databases, and assessing the impact on operational efficiency. The study identifies potential benefits, including improved accuracy, speed, and enhanced analysis capabilities. Additionally, it evaluates training needs, data exchange processes, and user acceptance for the successful adoption of the system.

# CHAPTER 4
# SYSTEM DESIGN AND ARCHITECTURE

## 4.1 System Design

The system architecture for our protein structure prediction model, employing LSTM, GNN, and TCN, encompasses key components tailored to our methodology. Initially, a data collection and pre-processing module is implemented to gather protein sequences from sources like the Protein Data Bank (PDB). This module addresses missing data and normalizes sequences to prepare them for subsequent processing. Feature extraction follows, where relevant features are derived from protein sequences utilizing techniques such as one-hot encoding, position-specific scoring matrices (PSSMs), or embeddings. Our approach integrates BLSTM, GNN, and BTCN into the model to capture both local and global dependencies. The training process involves optimizing the model using a suitable loss function. Our model is designed to dynamically consider important regions of the sequence. Evaluation metrics such as Root Mean Square Deviation (RMSD) are employed to assess the model's performance on test data.Continuous improvement and maintenance involve regular updates to the model using new data and incorporating advancements in BLSTM, GNN, and BTCN for enhanced accuracy in protein structure prediction.

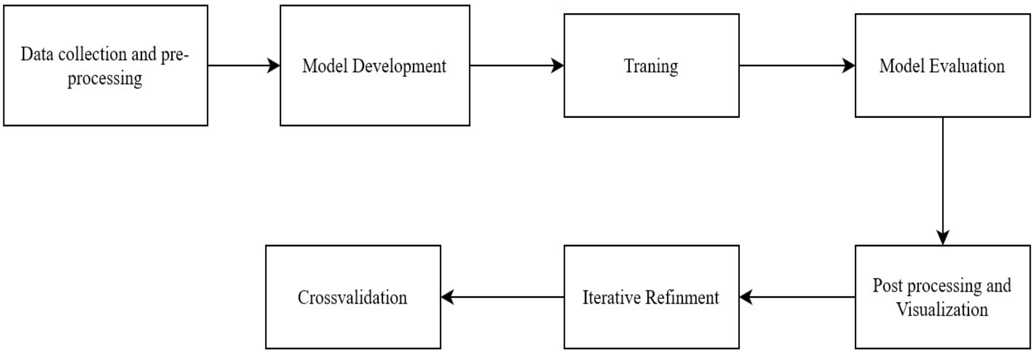

**Figure 4.1:** *System Design*

## 4.2 Use Case

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. We can explain and design our concept in a more structured way with the help of these diagrams. Use case Diagram for Protein

Structure Prediction is discussed below:

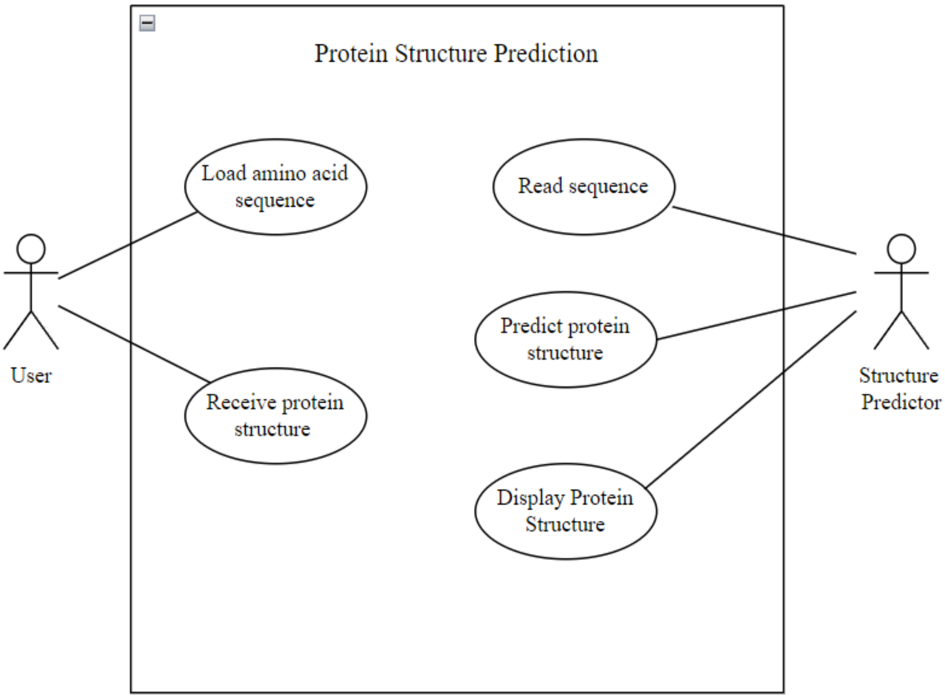
Protein Structure Prediction
Load amino acid sequence
Read sequence
User
Predict protein structure
Structure Predictor
Receive protein structure
Display Protein Structure

**Figure 4.2:** *Use Case Diagram*

# CHAPTER 5
# METHODOLOGY

## 5.1 Data collection and preprocessing

### 5.1.1 Data collection

For protein structure prediction, we leverage the PISCES dataset, a curated resource with pre-processed protein structures. This dataset serves as a valuable asset for training deep learning models in protein structure prediction tasks. By accessing and preprocessing the dataset, we can utilize the annotated information to enhance the accuracy of our predictions. The main dataset includes peptide sequences and their corresponding secondary structures. The PISCES server, drawing from the Protein Data Bank (PDB), generates lists of sequences based on chain-specific criteria and mutual sequence identity, widely employed in PSSP (Protein Secondary Structure Prediction) tasks. This streamlined data collection process provides a solid foundation for our research.

### 5.1.2 Data Pre-processing

Data pre-processing is a crucial step in protein structure prediction, particularly when utilizing the PISCES dataset. Upon accessing this valuable resource, the initial phase involves meticulous handling of the data to ensure its quality and consistency. This process includes addressing missing values, normalizing protein sequences, and implementing any necessary filtering or augmentation techniques. By rectifying inconsistencies and preparing the data with precision, the subsequent application of deep learning models for protein structure prediction is significantly bolstered. This emphasis on data quality lays a robust foundation, enhancing the overall effectiveness and reliability of the predictive models deployed in the intricate task of deciphering protein structures.

## 5.2 Model Development

### 5.2.1 BLSTM (Bidirectional Long Short Term Memory)

The Bidirectional Long Short-Term Memory (BLSTM) architecture is a sophisticated extension of the traditional LSTM, designed to enhance the model’s ability to capture bidirectional dependencies in sequential data. Comprising both forward and backward LSTM layers, the BLSTM processes input sequences in two directions concurrently—left-to-right and right-to-left. At each time step, the forward LSTM layer analyzes the input sequence, updating its hidden state and cell state. Simultaneously, the backward LSTM layer processes the sequence in reverse, capturing information from the future. The outputs from both directions are concatenated, creating a comprehensive representation that incorporates both past and future context for each time step. This bidirectional processing allows the BLSTM to excel in tasks where a nuanced understanding of temporal dependencies is crucial, such as speech recognition, natural language processing, and bio-informatics applications like protein structure prediction. The architecture's ability to effectively capture bidirectional dependencies makes it a powerful tool for tasks requiring a comprehensive analysis of sequential data.

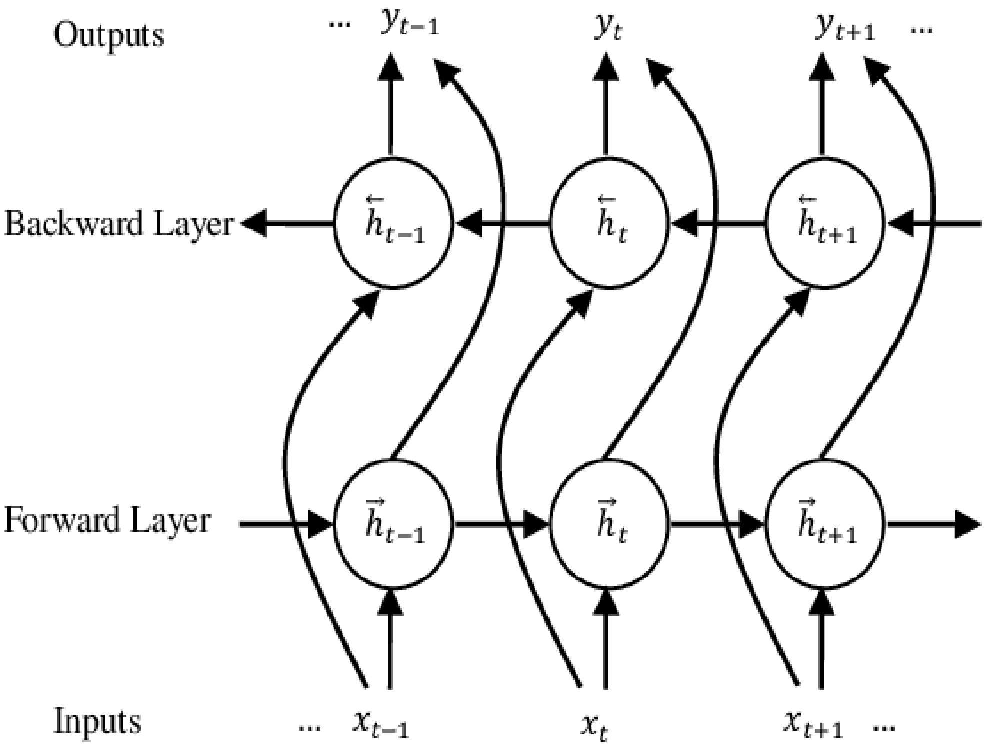

**Figure 5.1:** *Architecture of BLSTM*
*(Source:https://www.researchgate.net/figure/Architecture-of- BLSTM*$_{fig}1_3$07889752)

As shown in Figure, BLSTM consists of a forward LSTM and a backward LSTM. LSTM can automatically decide to discard unimportant information and retain useful information. For a standard LSTM cell at time $t$, the input feature is denotedas $x_t$, the output is denoted as $h_t$, and the cell state is denoted as $c_t$. Theforget gate $f_t$, the input gate $i_t$, and the output gate $o_t$ in the LSTMunit are calculated as follows:

$$f_t = \sigma(W_{xf} \times x_t + W_{hf} \times h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{xi} \times x_t + W_{hi} \times h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_{xo} \times x_t + W_{ho} \times h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \text{tanh}(W_{xc} \times x_t + W_{hc} \times h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \odot \text{tanh}(c_t) \quad (5)$$

where $\sigma$ is the sigmoid function, $W$ is the weight matrix, $b$ is the bias term, $\odot$ is the element-wise multiplication, and tanh is the hyperbolic tangent function.

### 5.2.2 BTCN (Bidirectional Temporal Convolution Network)

The Bidirectional Temporal Convolutional Network (BTCN) represents a dynamic advancement in neural network architectures tailored for sequential data processing. This innovative model integrates bidirectional temporal convolutions, combining the power of convolutional layers with the ability to capture dependencies in both past and future contexts. The architecture is structured with multiple layers of dilated convolutions, enabling an expansive receptive field and the efficient modeling of long-range dependencies. The bidirectional aspect ensures that information is gathered not only from preceding but also subsequent data points, enhancing the network's understanding of temporal relationships. The BTCN architecture excels in tasks where capturing intricate temporal patterns is crucial, such as in bioinformatics applications like protein structure prediction. Through its fusion of bidirectional processing and temporal convolutions, BTCN stands as a robust solution for tasks requiring a comprehensive analysis of sequential data.

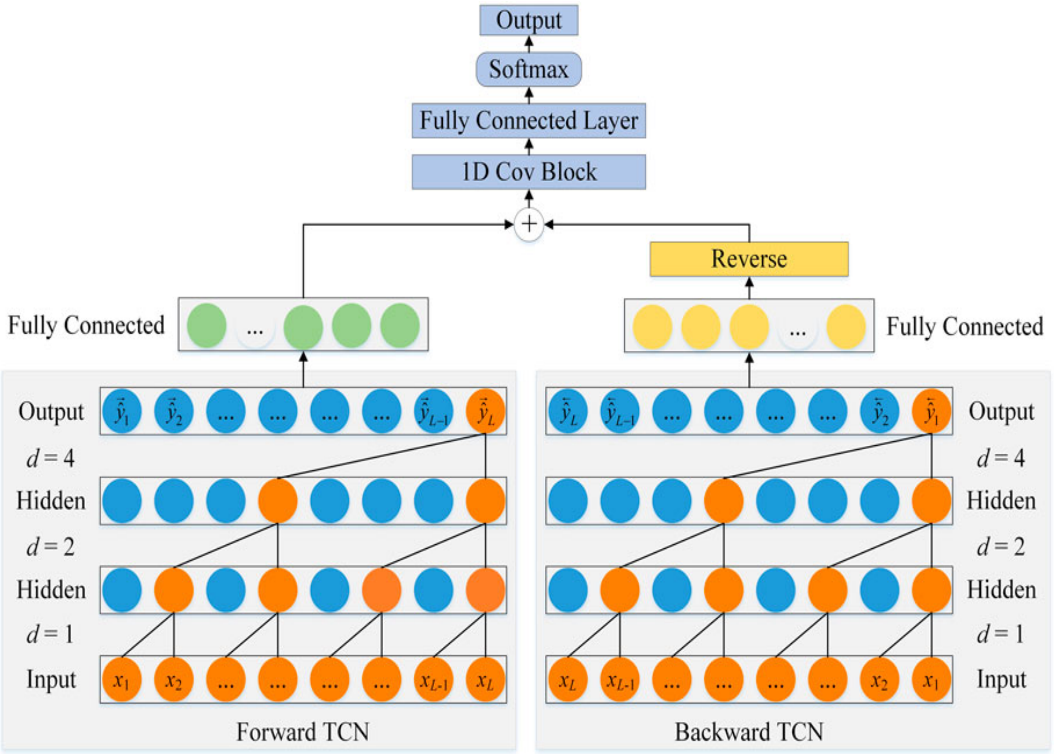
Output
Softmax
Fully Connected Layer
1D Cov Block
+
Reverse
Fully Connected
Fully Connected
Output
d = 4
Hidden
d = 2
Hidden
d = 1
Input
Forward TCN
Backward TCN
$x_1$ $x_2$ ... $x_{L-1}$ $x_L$
$x_L$ $x_{L-1}$ ... $x_2$ $x_1$
$\vec{y}_1$ $\vec{y}_2$ ... $\vec{y}_{L-1}$ $\vec{y}_L$
$\bar{y}_L$ $\bar{y}_{L-1}$ ... $\bar{y}_2$ $\bar{y}_1$

**Figure 5.2:** *Architecture of BTCN* *(Source:https://www.researchgate.net/figure/Architecture-of- BTCN*$_{fig}1_{3}07889752$)

As shown in Figure , the architecture of BTCN consists of forward TCN and backward TCN. Since the dilated causal convolution performs one-way operation on the sequence, we input the reverse sequence to the backward TCN for reverse feature extraction of the network. Letting $X$ denote a protein sequence, $L$ be the length of $X$, and $X =$

$\{x_1, x_2, ..., x_L\}$, $X^{\leftarrow}$ is the reverse sequence of $X$ defined as $X^{\leftarrow} = \{x_L, x_{L-1}, ..., x_1\}$. The Bidirectional Temporal Convolutional Network (BTCN) can be expressed as follows:

$$\hat{Y} = \text{TCN}^{\rightarrow}(X) \quad (8)$$

$$\hat{Y_1} = \text{TCN}^{\leftarrow}(X^{\leftarrow}) \quad (9)$$

$$\text{Output} = \text{softmax}\left(f\left(W\left(\text{1DCov}(\hat{Y} \oplus \hat{Y}_1^{\leftarrow})\right) + b\right)\right) \quad (10)$$

where $\text{TCN}^{\rightarrow}$ is the forward TCN with input as the forward sequence $X$, $\text{TCN}^{\leftarrow}$ is the backward TCN with input as the reverse sequence $X^{\leftarrow}$, $\oplus$ is the addition operation of matrices, $\hat{Y}$ and $\hat{Y_1}$ are the outputs of the forward and backward TCN, respectively, $\hat{Y}_1^{\leftarrow}$ is the reverse matrix of $\hat{Y_1}$, 1DCov is the 1D convolution operation of the residual block, $W$ and $b$ are the weight matrix and bias term of the fully connected layer, softmax is the activation function for classification, and Output is the final output of BTCN. The output $\hat{y}_t$ of the network at the current time $t$ can be determined by the input of the entire sequence, which is calculated as:

$$\hat{y}_t = \text{TCN}^{\rightarrow}(x_1, x_2, \ldots, x_t) \oplus \text{TCN}^{\leftarrow}(x_L, x_{L-1}, \ldots, x_t) \quad (11)$$

$$\text{Output}_t = \text{softmax}\left(f\left(W\left(\text{1DCov}(\hat{y}_t) + b\right)\right)\right) \quad (12)$$

We denote the forward $\text{TCN}^{\rightarrow}$ as TCN, and because the input of the backward $\text{TCN}^{\leftarrow}$ is the reverse sequence, it is also called reverse TCN and denoted as RTCN. Therefore, the architecture of the network is $BTCN = \text{1DCov(TCN} + \text{RTCN)}$, where 1DCov further optimizes the features. In the network, TCN operates on inputs at times $t$ and before $t$ $(x_1, x_2, \ldots, x_t)$, and RTCN operates on inputs at times $t$ and after $t$ $(x_L, x_{L-1}, \ldots, x_t)$. Therefore, the network can utilize bidirectional deep interactions to facilitate secondary structure recognition through forward and backward extraction of residue features. Furthermore, BTCN is not limited to PSSP; it applies to all sequences that require global semantics.

### 5.2.3 Graph Neural Network (GNN)

Graph Neural Networks (GNNs) have emerged as powerful tools in the realm of protein structure prediction, offering a paradigm shift from traditional sequential models. The architecture of a GNN revolves around its ability to model intricate relationships among amino acids by representing them as nodes in a graph. Edges between nodes signify interactions, capturing both local and global dependencies within the protein sequence. The GNN framework leverages graph convolutional layers, allowing each node to aggregate information from its neighbors, facilitating the integration of spatial dependencies. The message-passing mechanismenables GNNs to discern complex patterns and relationships, crucial for understanding the nuanced folding patterns of proteins. Additionally, GNNs are inherently capable of handling irregular graph structures, making them well-suited for capturing the diverse and dynamic nature of amino acid interactions. By combining graph-based representations with deep learning techniques, GNNs offer a holistic approach to protein structure prediction, promising improved accuracy and efficiency in capturing the intricate spatial dependencies critical for unraveling the three-dimensional conformation of proteins.

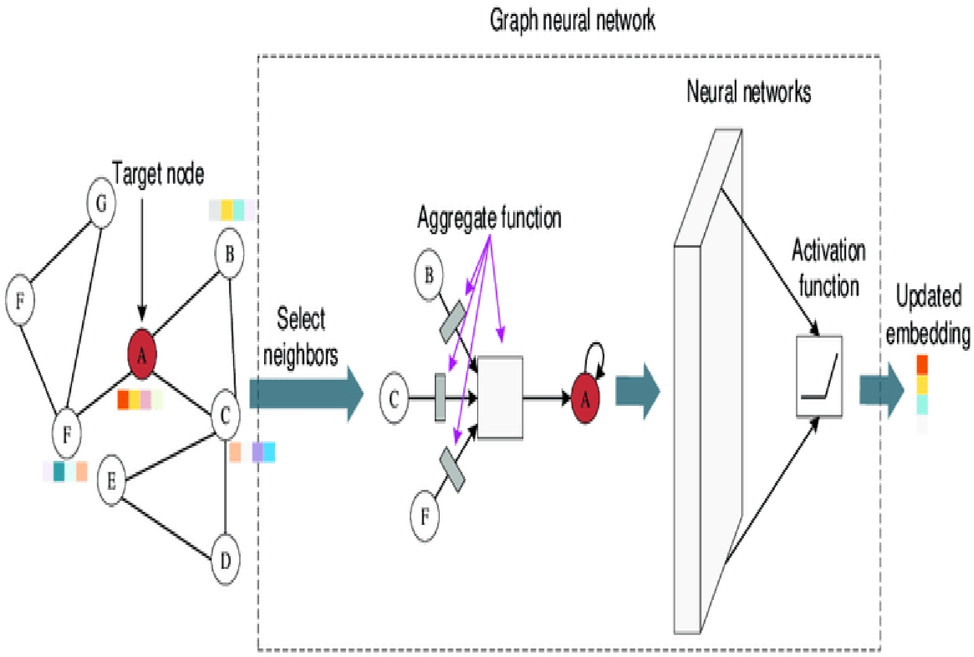

**Figure 5.3:** *Architecture of GNN (Source:https://www.researchgate.net/figure/A-typical-and-basic-architecture-and-processing-procedures-of-GNN-First-GNN-selects*$_fig2_352526255))$

In the context of Graph Neural Networks (GNNs) for protein secondary structure prediction, the model's functioning involves several key components expressed through mathematical equations. The aggregate function ($h_{\mathcal{N}(v)}$) captures information from neighboring nodes, often employing mean aggregation. The neural network layer ($W$, $b$) transforms node features, where $a_v = W \cdot h_v + b$. The activation function, in this case, is the rectified linear unit (ReLU), introducing non-linearity and leading to $h'_v = \text{ReLU}(a_v)$. The update embedding process combines the transformed features and aggregated information, often employing concatenation and another neural network layer: $h_v = \text{ReLU}(W' \cdot [h'_v, h_{\mathcal{N}(v)}] + b')$. This iterative process across multiple layers captures complex dependencies within the protein graph, allowing the GNN to learn and predict protein secondary structures. Training involves optimizing parameters ($W$, $b$, $W'$, $b'$) to minimize the difference between predicted and actual secondary structures in the training data, demonstrating the GNN’s adaptability to graph-structured biological data.

# CHAPTER 6
# WORK PROGRESS

## 6.1 Work Done

### 6.1.1 Research

In our pursuit of ensuring the success of our project, we conducted extensive research on protein structure prediction tasks and related bioinformatics concepts. This thorough investigation has provided us with a profound understanding of the intricacies involved in predicting protein structures, as well as insight into various techniques employed in the field. After careful consideration of numerous existing datasets, we have identified the PISCES dataset as the most suitable for our project. This dataset offers protein sequences with corresponding structural labels, specifically designed for protein structure prediction tasks. Moreover, we have acknowledged the significance of incorporating biochemical concepts into our project, recognizing their primary relation to biology and bioinformatics. This realization has emphasized the need for a dedicated research phase and the utilization of various biochemical concepts, which has become a key focus in our project.

### 6.1.2 Data Collection

One of the initial steps in our project was to collect dataset, which contains protein sequences and their corresponding structures label. This dataset has been specifically designed for protein structure prediction tasks, making it an ideal resource for our project. By acquiring this dataset, we have obtained the necessary foundation to begin our analysis and modeling. The data within the PISCES dataset will serve as the basis for our research and will enable us to develop a robust and accurate protein structure prediction model.

### 6.1.3 Visualization

In the visualization aspect of our project, we employed diverse techniques to gain comprehensive insights into the PISCES dataset. Utilizing histograms, we successfully identified the maximum sequence length of amino acids and their corresponding frequencies, offering valuable information about the dataset's distribution. Beyond histograms, we further enriched our visual exploration by showcasing the proportion of each amino acid in a tabular format. This additional representation allowed for a detailed examination of the composition of amino acids within the dataset. Additionally, we extended our visualization efforts to illustrate the proportion of sequences containing non-standard amino acids, providing a nuanced perspective on the dataset's complexity. These visualizations, spanning sequence length distribution, amino acid proportions, and the prevalence of non-standard amino acids, collectively facilitated a deeper understanding of the dataset's characteristics. Such insights proved instrumental in guiding subsequent stages of our project, aiding in informed decision-making and enhancing the overall efficacy of our protein structure prediction endeavor.

### 6.1.4 Data Preprocessing

In the data preprocessing phase for the PISCES dataset, our approach focused on ensuring the dataset's quality and suitability for protein structure prediction. We systematically removed irrelevant information and addressed missing values to narrow our analysis to essential data. Standardizing the data format for consistency streamlined subsequent stages of the project. For numerical representation of protein sequences, a key step involved employing one-hot encoding, transforming each amino acid into a binary vector of length 20. This encoding method effectively captured the categorical information, facilitating seamless incorporation into our analyses and modeling. Concurrently, we generated n-grams from sequences to capture local patterns. Additionally, we tokenized input sequences using one-hot encoding and target sequences using character-level encoding, providing comprehensive coverage of sequence information. These preprocessing techniques, including n-gram generation and encoding methods, ensured the PISCES dataset's readiness for feature extraction and subsequent model development. These steps collectively enhanced the dataset's quality and reliability, contributing to improved accuracy in protein structure predictions.

### 6.1.5 Data Split

In our project, a systematic and robust approach was undertaken for data splitting, crucial for training, validating, and evaluating the performance of our protein structure prediction model. We utilized the `train_test_split` function to effectively partition our dataset into training and testing sets. Specifically, we divided both the input data and its corresponding target data into training and testing subsets, ensuring a representative distribution. The data splitting process involved allocating 60% of the dataset for training and 40% for testing, providing a balanced distribution to enhance the model's learning and generalization capabilities. Additionally, a random seed (`random_state=0`) was set to maintain reproducibility across experiments. This strategic data splitting methodology enabled us to train our model on a diverse set of sequences while validating its performance on an independent subset, contributing to the robustness and reliability of our protein structure prediction model.

### 6.1.6 Model Development

In our project, a significant focus has been dedicated to the development of robust and effective models for predicting protein structure labels. Two key models, namely Bidirectional Long Short-Term Memory (LSTM) and Bidirectional Temporal Convolutional Network (TCN), were meticulously designed and implemented. The Bidirectional LSTM, known for its ability to capture long-range dependencies in sequential data, played a pivotal role in our model architecture. This architecture enables the model to consider information from both past and future sequences, enhancing its understanding of complex relationships within protein sequences. Additionally, the incorporation of the Bidirectional TCN further enriched our model's capabilities. TCNs are adept at capturing hierarchical features and have been proven effective in various sequence-based tasks. By adopting a bidirectional approach in both LSTM and TCN components, our model benefits from a comprehensive analysis of the input data, contributing to its overall predictive accuracy. The development of these sophisticated models reflects our commitment to leveraging state-of-the-art techniques in bioinformatics to advance the field of protein structure prediction.

### 6.1.7 Model Training

The training phase of our dataset using the developed models, which encompasses the meticulously designed Bidirectional Long Short-Term Memory (LSTM) and Bidirectional Temporal Convolutional Network (TCN), we embarked on a comprehensive journey to harness the predictive potential of our models. The training commenced by feeding sequences of amino acids into the models, initializing their parameters for an effective learning process. Leveraging gradient descent-based algorithms, specifically employing backpropagation, we optimized the models' parameters iteratively to minimize prediction errors and enhance their predictive accuracy. To expedite the training process, we judiciously applied optimization techniques, such as Adam or RMSprop, which played a crucial role in navigating the complex parameter space. These optimization techniques facilitated quicker convergence and enabled our models to efficiently capture the intricate patterns and dependencies within the training datasets. The validation data split served as a valuable checkpoint, guiding the adjustment of hyperparameters to strike a balance between model complexity and generalization. This rigorous training methodology underscores our commitment to developing models that are not only sophisticated in their architecture but also finely tuned to make accurate predictions on diverse and unseen protein sequences.

## 6.2 Work Remaining

### 6.2.1 Model Development

Having successfully developed the Bidirectional Long Short-Term Memory (BLSTM) and Bidirectional Temporal Convolutional Network (BTCN) models for protein structure prediction in the earlier phases of our project, we are now advancing to the development phase of a Graph Neural Network (GNN) model. This transition marks a strategic expansion of our modeling approach, leveraging the unique capabilities of GNNs to capture intricate relationships and dependencies in protein sequences. The GNN model, designed to operate on graph-structured data, holds promise in enhancing our understanding of complex interactions within amino acid sequences. This development phase involves crafting an architecture that integrates graph-based learning, allowing the model to effectively learn from the inherent graph structure present in protein sequences. As we navigate this frontier in model development, our goal remains to further elevate the predictive accuracy and comprehensiveness of our protein structure prediction endeavors, contributing to advancements in the field of bioinformatics.

### 6.2.2 Model Evaluation

Evaluate the trained model's performance using the validation datasets. Use metrics like accuracy, precision, recall, and F1 score to assess its effectiveness. Additionally, employ structural metrics like RMSD to evaluate the model's ability to predict accurate protein structures.

### 6.2.3 Hyperparameter Tuning

In the pursuit of refining our protein structure prediction models, a pivotal step is hyperparameter tuning to enhance overall performance. This process involves meticulous adjustments to parameters such as learning rate, batch size, and the number of layers or hidden units within the Bidirectional Long Short-Term Memory (BLSTM), Bidirectional Temporal Convolutional Network (BTCN), and Graph Neural Network (GNN) models. Leveraging techniques like grid search or random search, we systematically explore the hyperparameter space to identify optimal configurations. By fine-tuning these critical parameters, we aim to strike an optimal balance that maximizes the models' learning capabilities and predictive accuracy. This iterative optimization process ensures that our models are finely calibrated to extract the most relevant features from protein sequences, ultimately advancing the accuracy and reliability of our protein structure predictions.

### 6.2.4 Model Testing

Test the final trained model on the testing dataset to evaluate its generalization ability. This step determines how well the model performs on unseen protein sequences from the PISCES dataset.

### 6.2.5 Cross-Validation

Cross-validation is a vital step in our project, ensuring robust model assessment by partitioning the dataset into multiple subsets. Techniques like k-fold cross-validation systematically train and validate the model on different folds, enhancing its generalization and reliability across diverse data samples. This approach aids in mitigating over-fitting and provides a comprehensive evaluation of our protein structure prediction models.

### 6.2.6 Performance Analysis

Analyze the model's performance and compare it with existing methods or benchmarks. This assessment helps understand the effectiveness and efficiency of the proposed BLSTM,BTCN and GNN model for protein structure prediction using the PISCES datasets.

# REFERENCES

1. Hartmanis, M. (2019). *Protein Structure and Function*. In Biochemistry (pp. 932). Springer.
2. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). *Proteins*. In Molecular Biology of the Cell (4th edition). Garland Science.
3. Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., …Kohli, P. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792), 706-710.
4. Cao, R., Freitas, C., Chan, L., Sun, M., Jiang, H., & Chen, Z. (2017). *ProLanGO: Protein Function Prediction Using Neural Machine Translation Based on a Recurrent Neural Network. Molecules*, 22(10), 1732.
5. Greener JG, Kandathil SM, Jones DT. (2021). Link: `https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008990`
6. King, J. E., & Koes, D. R. (2021). *SidechainNet: An all-atom protein structure dataset for machine learning. Proteins: Structure, Function, and Bioinformatics*, 89(11), 1489-1496. doi: 10.1002/prot.26169
7. Alquraishi, M. (2019). *End-to-end differentiable learning of protein structure. PLoS Computational Biology*, 15(11), e1007449. doi: 10.1371/journal.pcbi.1007449.
8. Baldi, Pierre, and Gianluca Pollastri. ”The principled design of large-scale recursive neural network architectures–dag-rnns and the protein structure prediction problem.” *The Journal of Machine Learning Research*, 4 (2003): 575-602.
9. H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, ”Low illumination underwater light field images reconstruction using deep convolutional neural networks,” *Future Gener. Comput. Syst.*, vol. 82, pp. 142–148, May 2018.
10. C. Chen, L. Chen, X. Zou, and P. Cai, ”Prediction of protein secondary structure content by using the concept of Chou’s pseudo amino acid composition and support vector machine,” *Protein Peptide Lett.*, vol. 16, no. 1, pp. 27–31, Jan. 2009.
11. M. N. Islam, S. Iqbal, A. R. Katebi, and M. T. Hoque, ”A balanced secondary structure predictor,” *J. Theor. Biol.*, vol. 389, pp. 60–71, Jan. 2016.
12. B. Yang, ”Predicting protein secondary structure using a mixed-modal SVM method

in a compound pyramid model," *Knowl.-Based Syst.*, vol. 24, no. 2, pp. 304–313, 2011.

13. N. P. Bidargaddi, M. Chetty, and J. Kamruzzaman, "Combining segmental semi-Markov models with neural networks for protein secondary structure prediction," *Neurocomputing*, vol. 72, nos. 16–18, pp. 3943–3950, Oct. 2009.
14. X.-Q. Yao, H. Zhu, and Z.-S. She, "A dynamic Bayesian network approach to protein secondary structure prediction," *BMC Bioinf.*, vol. 9, no. 1, p. 49, Dec. 2008.
15. S. A. Malekpour, S. Naghizadeh, H. Pezeshk, M. Sadeghi, and C. Eslahchi, "Protein secondary structure prediction using three neural networks and a segmental semi-Markov model," *Math. Biosciences*, vol. 217, no. 2, pp. 145–150, Feb. 2009.
16. Y. T. Tan and B. A. Rosdi, "FPGA-based hardware accelerator for the prediction of protein secondary class via fuzzy K-nearest neighbors with Lempel–Ziv complexity based distance measure," *Neurocomputing*, vol. 148, pp. 409–419, Jan. 2015.
17. S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, no. 1, pp. 1–11, May 2016.
18. Y. Liu and J. Cheng, "Protein secondary structure prediction based on wavelets and 2D convolutional neural network," in *Proc. 7th Int. Conf. Comput. Syst.-Biol. Bioinf.*, Dec. 2016, pp. 53–57.
19. C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, 2004, pp. 32–36.
20. Z. Li and Y. Yu, "Protein secondary structure prediction using cascaded convolutional and recurrent neural networks," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 1–8.
21. R. Heffernan, Y. Yang, K. Paliwal, and Y. Zhou, "Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility," *Bioinformatics*, vol. 33, no. 18, pp. 2842–2849, Sep. 2017.
22. Y. Guo, W. Li, B. Wang, H. Liu, and D. Zhou, "DeepACLSTM: Deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction," *BMC Bioinf.*, vol. 20, no. 1, p. 341, Dec. 2019, doi: 10.1186/s12859-019-2940-0.