PSSRcomp: A detailed analysis of secondary protein structure prediction

by

Kaushik Datta
23341060
MD. Shafiul Alam
20301262

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Kaushik Datta	MD. Shafiul Alam
23341060	20301262

Approval

The thesis/project titled "PSSRcomp: A detailed analysis of secondary protein structure prediction" submitted by

1. Kaushik Datta (23341060)
2. MD.Shafiul Alam (20301262)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 3, 2025.

Examining Committee:

Supervisor:

Dr. Md. Ashraful Alam
Associate Professor
Computer Science Engineering
BRAC University

Cosupervisor:

Dr. Md. Golam Rabiul Alam
Professor
Computer Science Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

The field of protein structure research has seen a significant transformation with the introduction of deep learning. Based on protein primary structure sequences, predicting secondary structures plays a crucial role in determining and understanding the roles of an amino acid sequence. Protein structural analysis was traditionally performed using time-consuming and sometimes imprecise technologies. We composed four types of prebuilt models - LSTM, RNN, GNN, ANN, etc, to predict secondary structures from different latest datasets of PDBbind and PISCES to understand the learning process HMM, Sequence embedding, and PSSM in a broader sense of machine readability. The q3 and q8 states of the secondary sequence are measured and analyzed for their distribution throughout the data set to comprehend their structural properties. In the post-AlphaFold era, these amino acid sequence data were theoretically capped at 94% accuracy. Our research hyper-optimized those approaches to find effective sequence prediction methods for better accuracy. Our optimized model acquired 0.9031 accuracy with impressive test results and custom evaluation metrics. This integration facilitates the relationship between protein structural properties to identify differences between protein learning models. Personalized treatment, drug development, and a better understanding of illnesses have all been made possible by the capacity to predict protein structures using these cutting-edge deep learning approaches reliably. However, it is important to use these computational models as a backup. Human expertise and domain knowledge are essential to interpret and validate data.

Keywords: Deep learning, neural network graph, sequence integration, prediction, fusion layer, protein classification, protein structures, structural similarities, ANN, RNN, LSTM.

Table of Contents

List of Figures

Chapter 1

Introduction

Protein structure predictions have long been a challenging issue in bioinformatics, but deep learning techniques have significantly improved accuracy and efficiency. It is essential to emphasize that protein structure prediction using deep learning remains an active research area with many challenges ahead. A major complication is the intricate nature of protein structures resulting from the amino acid sequence[1]. Accurately predicting these structures based solely on sequence information is still incredibly difficult.While many proteins are known, there is a scarcity of high-confidence experimental evidence for protein architectures. Evaluating the accuracy and reliability of predicted protein structures presents another significant obstacle. Experimental validation methods, such as cryo-electron microscopy and X-ray crystallography, are both time-consuming and labor-intensive, and computational predictions that lack solid validation can be misleading. Although deep learning methods are advancing in protein structure prediction, they are not yet capable of delivering dependable predictions across various proteins or complex structural motifs. Despite the potential benefits of progress in deep learning methods, they also encounter limitations such as data scarcity, model transferability, and challenges related to empirical validation. Their functionality is heavily dependent on their topology, especially the secondary structure (H = alpha-helices, B = beta-sheets, coil). Predicting secondary structures is a crucial task in computational biology and bioinformatics, as it provides insights into protein folding, stability, and interactions. Traditional experimental methods like X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy yield accurate structural determinations, but they can also be costly and time-consuming.

As a result, computational approaches have gained prominence in secondary structure prediction by leveraging improvements in AI and ML to produce accurate and efficient secondary structure information.With protein sequence information now readily available in databases such as UniProt and Protein Data Bank (PDB), the performance of machine learning-based techniques for predicting secondary structure has improved significantly. Over the past few years, deep learning approaches such as Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Graph Neural Networks (GNNs) have made statistical methods and position-specific scoring matrices (PSSMs) obsolete. Such models are capable of capturing complex patterns present in the sequence and predicting structural features more accurately [3]. In this work, we investigate secondary protein structure prediction through several deep-learning architectures. The

main purpose here is to evaluate the impact of different encoding methods, model architectures, and hyperparameter fine-tuning on predictive accuracy. Several computational approaches are utilized in the study, including sequence encoding, where the amino acid sequence is converted to a sequence of numerical values by applying binary encoding and padding techniques. The encoded data is used as an input to deep learning models to effectively extract structural features and predict secondary structures.

Our study spans a variety of architectures, from ANN to RNN, LSTM, and GNN, each leveraging a different approach to sequence dependency capture. ANNs are the basic model of deep learning, which uses feed-forward connections and sigmoid activation functions. RNNs use recurrent connections to store information from prior sequence locations, allowing them to be effective at modeling sequential data. Long Short-Term Memory (LSTM) is an advanced sequence modeling approach that employs memory gates to retain long-range dependencies in protein sequences, adding significant value to structural predictions of complex proteins. Lastly, GNNs incorporate graph-based representations to model the residue relations between sequences, providing a different view in the secondary structure prediction method. We assess model performance through accuracy in predictions for Q3 (Coil, Helix, Beta-Sheet) and Q8 (eight-class secondary structure) prediction accuracy, which is trained by categorical-cross entropy loss functions through O(1) Adam and RMSprop optimizers. All models are evaluated with standard measures (training/validation accuracy, mean absolute error (MAE), root mean squared error (RMSE), and relative absolute error (RAE)) after they are trained and verified with vast datasets. These metrics help in determining both the individual accuracies of each model in identifying secondary structural elements as well as their predictive power.

The results of our study highlight the comparative strengths and weaknesses of each model. While ANNs provide a simple yet effective baseline, RNNs and LSTMs outperform them by capturing long-term dependencies in sequence data. GNNs, on the other hand, introduce a new paradigm by representing sequences as graphs and using n-gram features to enhance predictive capabilities. Our findings provide valuable insights into the future of protein structure prediction, offering a roadmap for selecting the most suitable architectures based on specific dataset characteristics and application needs. The paper initially provides an in-depth discussion of dataset preparation and encoding techniques. It also elaborates on the architectures and training methodologies used for secondary structure prediction. The study also presents the experimental results, including accuracy comparisons, error metrics, and confusion matrices. Moreover, discusses the implications of our findings and the potential for further improvements in deep learning-based secondary structure prediction. Finally, concludes the paper, summarizing key takeaways and future directions in the field of protein bioinformatics. By assessing deep learning approaches and offering insights that might aid in the creation of more reliable and accurate computer models, this thorough investigation seeks to further the area of protein structure prediction.

1.1 Research Objectives

This thesis aims to develop a deep understanding of secondary protein sequence prediction using pre-trained models. Thus models like - LSTM, GNN, ANN, and RNN each have a different approach to identifying the unique elements of a primary protein structure. Building correlation for different protein prediction models to get better performance is the main goal. In the data preparation process several types of feature engineering like - PSSM, Sequence embedding, Orthogonal Encoding, One-Hot encoding, and HMM, etc., will be used for thorough analysis for improved accuracy and confidence score. To analyze amino acid sequences and capture long-range dependencies, implement a multi-model classification approach to enhance prediction accuracy by leveraging diverse model strengths, address data quality and diversity challenges by utilizing high-quality experimental datasets and robust preprocessing techniques, and validate the model's performance against established benchmarks and experimental data to ensure reliability and accuracy.

1.2 Report Organization

This report is organized into several key sections to systematically analyze and present the research on protein structure prediction using deep learning techniques, specifically focusing on LSTM, RNN, GNN, and ANN. The introduction provides a comprehensive background on the significance of protein structure prediction and the revolutionary impact of computational methods in these fields. The literature review delves into the advancements and challenges of using deep learning models, highlighting significant contributions such as AlphaFold and the potential of multi-model classification approaches. Following this, the methodology section details the proposed prediction model, emphasizing the integration of different feature engineering to capture complex spatial relationships and enhance prediction accuracy. The results and discussion sections present the findings from our experimental studies, comparing the performance of various models and discussing the implications of our approach. Finally, the conclusion summarizes the key insights, acknowledges the limitations, and suggests directions for future research. Each section is designed to build upon the previous one, providing a cohesive and thorough analysis of the subject matter.

1.3 Protein Structure

The structures are closely related, influencing their interactions with other molecules and roles in biological processes. Proteins are mainly chains of amino acid sequences. There are more than twenty amino acids that sequentially create chains that build up protein structure. Understanding protein structure makes it simpler to identify the underlying causes of illnesses and develop efficient treatment plans. This includes structural bioinformatics, biochemistry, molecular biology, and pharmaceutical development. Primary, secondary, tertiary, and quaternary structures are the four levels into which proteins are arranged hierarchically. Every level contributes differently to the overall stability and functionality of the protein [4].

1.3.1 Primary Structure

A protein's distinct arrangement of amino acids in a linear polypeptide chain is referred to as its fundamental structure. Peptide bonds bind these amino acids together to form a continuous backbone. The appropriate gene sequence determines the fundamental structure, which in turn determines the final form and function of the protein. Serious functional alterations can arise from even a single mutation in this region, occasionally leading to illnesses like sickle cell anaemia, which is brought on by a single amino acid substitution in the haemoglobin protein. [6].

1.3.2 Secondary Structure

The secondary structure arises due to local folding patterns within the polypeptide chain, primarily stabilized by hydrogen bonding between backbone atoms. The two most common secondary structural motifs are:

- **Alpha-Helices (α-Helices)** Alpha-helices are right-handed coils that are often found in membrane proteins and enzymes. They are stabilized by hydrogen bonds between every fourth amino acid, ensuring both structural stiffness and flexibility.
- **Beta-Sheets (β-Sheets)** Beta-sheets consist of laterally organized beta-strands held together by interstrand hydrogen bonds. The orientation of the strands determines whether the sheets are parallel or antiparallel. Beta-sheets are frequently observed in enzyme active sites and fibrous proteins.

Understanding secondary structures is critical for protein function prediction, as they serve as fundamental building blocks in larger structural formations.
The tertiary structure represents the overall three-dimensional folding of a single polypeptide chain, which determines the protein's biological function. It arises from interactions among secondary structure elements and is stabilized by various forces, including:

- **Hydrogen Bonds** (between polar groups)
- **Disulfide Bridges** (covalent bonds between cysteine residues)
- **Hydrophobic Interactions** (between non-polar amino acids)
- **Ionic Bonds** (between charged side chains)

1.3.3 Computational Approaches to Protein Sequence Prediction

Conventional experimental techniques, such as cryo-electron microscopy, NMR spectroscopy, and X-ray crystallography, are very accurate yet expensive and time-consuming. Computational methods, especially machine learning and deep learning models, have therefore become strong substitutes. By effectively predicting secondary structures from protein sequence data, these techniques improve structural insights while drastically lowering experimental effort. Protein structure is arranged hierarchically, which greatly influences biological activity [19]. In computational biology, secondary structure prediction is an essential step that connects primary sequencing data with three-dimensional protein modeling. By using artificial intelligence, neural networks, and graph-based methods, researchers hope to increase the accuracy of protein structure predictions, potentially leading to significant advancements in biotechnology, medicine, and bioinformatics.

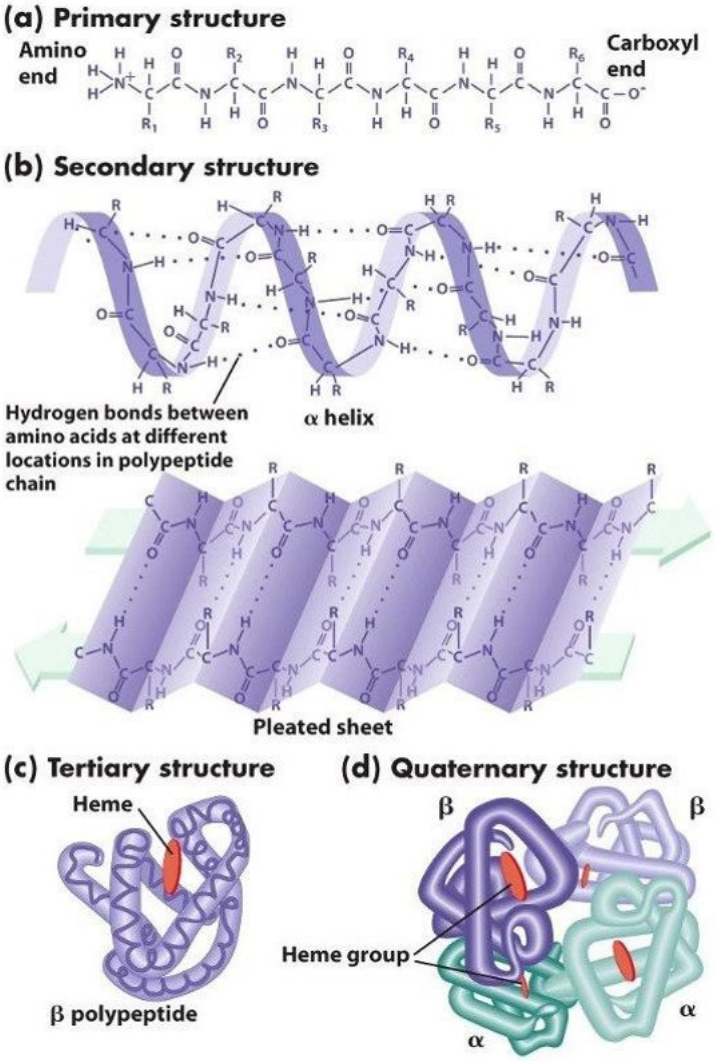


Figure 1.1: Protein structure

Proteins are made of building blocks called amino acids, the sequence of which determines the structure and function of proteins. A linear arrangement of peptide

bonds holding together a sequence of amino acids is called a protein, and once folded into its three-dimensional form, these are known as polypeptides. The biological function of the protein is determined by the properties and interactions of these amino acids[11].
Each amino acid consists of a central carbon (α-carbon) bonded to:

- An amino group (-NH)
- A carboxyl group (-COOH)
- A hydrogen atom (H)
- A variable side chain (R-group) that determines the chemical properties of the amino acid

The R-group is unique for each of the 20 standard amino acids and influences their behavior in protein structures.
Amino acids can be classified based on their R-groups into the following categories:
Hydrophobic (Non-polar) Amino Acids: These amino acids have non-polar side chains and are typically found in the interior of proteins, away from water. Examples include:

- Glycine (Gly, G) – The simplest amino acid, with a hydrogen R-group
- Alanine (Ala, A) – Has a methyl (-CH) side chain
- Leucine (Leu, L), Isoleucine (Ile, I), Valine (Val, V) – Branched-chain amino acids
- Phenylalanine (Phe, F), Tryptophan (Trp, W), Tyrosine (Tyr, Y) – Aromatic amino acids
- Methionine (Met, M) – Contains sulfur and is essential for protein synthesis

Hydrophilic (Polar) Amino Acids: These amino acids have polar side chains and are often found on the surface of proteins, interacting with water. Examples include:

- Serine (Ser, S), Threonine (Thr, T) – Contain hydroxyl (-OH) groups
- Asparagine (Asn, N), Glutamine (Gln, Q) – Contain amide (-CONH) groups
- Cysteine (Cys, C) – Contains a thiol (-SH) group and can form disulfide bonds, stabilizing protein structure

Acidic (Negatively Charged) Amino Acids: These amino acids have carboxyl (-COO) groups, making them negatively charged at physiological pH:

- Aspartic acid (Asp, D)
- Glutamic acid (Glu, E)

Basic (Positively Charged) Amino Acids: These amino acids have positively charged side chains at physiological pH:

- Lysine (Lys, K) – Contains an amino (-NH) group
- Arginine (Arg, R) – Has a guanidinium (-C(NH)) group
- Histidine (His, H) – Contains an imidazole ring and plays a role in enzyme activity

Peptide bonds link amino acids together, forming through a condensation reaction (dehydration synthesis) between the carboxyl group of one amino acid and the amino group of another. By joining these two (or more) homologous building blocks, there is the release of a molecule of water (HO) and a linear polypeptide chain.

The Importance of Amino Acids in Protein Activity: Catalysis – Hundreds of enzymes contain specific amino acids in their active sites (such as proteases and kinases). Structural Support – Collagen, for example, provides mechanical strength. Transport & Storage – Hemoglobin carries oxygen, while ferritin stores iron. Hormonal Regulation – Insulin plays a key role in glucose metabolism. Immune Defense – Antibodies help defend against pathogens.
Mutations that change the sequence of amino acids can also affect protein function. Some examples include:

- Sickle Cell Anemia – One amino acid change (glutamic acid to valine) in hemoglobin changes the shape of red blood cells.
- Cystic Fibrosis – Caused by a deletion mutation affecting a protein that transports chloride ions.

Proteins are essentially their amino acid sequences, which dictate protein function as well as protein structure. Their precise order, determined by genetic instructions, guarantees that they fold correctly and remain biologically active. Knowledge of amino acids is important in various disciplines like medicine, genetics, and biotechnology, since even minor alterations in these sequences can lead to disease or health issues.

Chapter 2

Literature Review

Protein structure prediction is currently one of the most important areas of computational biology, especially after the introduction of deep learning methods. Data collection and cleanup is the first and essential step in every deep learning approach for protein structure prediction. Protein Data Bank (PDB) databases, which host verified protein structures obtained from experimental studies, and UniProt, a complete database of protein sequences, are frequently utilized to curate datasets for protein structure prediction. The data extraction process includes amino acid sequences with their corresponding 3D structures, and these data act as the training base for the deep learning model [13]. The robustness and generalization capability of these techniques are heavily reliant on the quality and quantity of the datasets used. The research delivers a comprehensive examination on contrasting feature extraction strategies and deep learning architectures, specifically, Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and lastly Graph Neural Networks (GNN). Their importance in hyperparameter tuning, training and evaluation approaches are explained, and a comparative analysis of performances of these models is disclosed.

It emphasizes the significance of choosing the right feature extraction techniques to enhance the accuracy of protein structure predictions while also streamlining computational costs. Results evaluation and comparative analysis aims to explore strengths and weaknesses of each protein structure prediction model[22]. In this study, we systematically investigate the impact of various deep learning architectures including PFs and CNNs on protein structure prediction. The project is categorized into several significant phases including data preparation, feature extraction, model selection, hyperparameter tuning, training, evaluation, and comparison. These steps are very important, however, they all become relative when it comes to using the models to solve the various tasks of predicting the protein structure [23].

In any deep learning-based protein structure prediction pipeline, data preprocessing represents the first step. As the science progresses, protein structure prediction is a common subject of active research in computational biology, and new models and methods are constantly exploring what is possible. This work provides a comprehensive evaluation of deep learning architectures for protein structure prediction. Combining data preparation, feature extraction, model selection, and evaluation to define the best models for a range of protein structure prediction problems. This

systematic approach guarantees speed, reliability, and repetition in research within the domains of computational biology and structural bioinformatics. Once the data is collected, feature extraction steps in as the next phase to convert the raw sequence data into a format understandable to machine learning models [24]. Feature Extraction is performed using techniques such as One-Hot Encoding (OH), Hidden Markov Models (HMM), embeddings, Position-Specific Scoring Matrices (PSSM), and Graph Orthogonal Encoding in Protein Structure prediction. One-hot encoding encodes each amino acid as a vector, this does not take into account complex dependencies and relationships between amino acids. However, while HMM-based approaches effectively model sequential dependencies, such approaches have been predominantly used for aligning protein sequences, and the probabilistic nature of HMMs does not lend themselves as naturally to understanding proteins as families. Embedding techniques, similar to how AlphaFold's embeddings, encode protein sequences capture semantic relationships of amino acid As an added bonus, PSSMs also provide biological phylogenetic information, allowing for the scoring of amino acid positions based on their probability, potentially improving the understanding of protein function and structure. And finally, Graph and Orthogonal Encoding – which uses the power of Graph Neural Networks (GNNs) to formulate proteins as graphs: Each residue corresponds to a node, while edges capture interactions between residues, preserving spatial and relational information that is crucial for constructing protein structures.

Another major part of data preparation is handling missing values. Common techniques to impute missing values include expectation-maximization (EM) algorithms and k-nearest neighbour(KNN) imputation. Data Augmentation is another important preprocessing method to use in deep learning approaches for predicting protein structure. Diversity enhances the generalizability of the model, which can be achieved by introducing simulated mutations, generating synthetic sequences, and employing sliding window techniques to augment diversity in training datasets [26]. Introduction to feature extraction for protein sequences: As deep learning models require input data in a structured numeric format, feature extraction is a critical process in transforming raw protein sequences into meaningful representations that can be understood by the model. The feature extraction method used to capture this information plays a key role in the overall performance of protein structure prediction models. Position Specific Scoring Matrices (PSSM) are a common method to extract evolutional information contained in the protein sequences. PSSM encodes the frequency of each amino acid appearing at a particular position in a given sequence, derived from alignments of homologous proteins of known sequence. Another commonly used method is physicochemical property encoding, which encodes amino acid sequences according to properties such as hydrophobicity, charge and molecular weight. This allows the model to more effectively leverage functional and structural similarities between proteins.

Graph-based representations are increasingly being utilized for protein structure modelling. Graph Nodes and Edges: GNNs can represent amino acid interactions as succinct nodes and edges capturing long range dependencies and spatial relationships in protein sequences [27]. Moreover, the usage of structural embeddings from 3D protein conformation models enhances the predictive power of deep learning

architectures in predicting secondary. Today, there are a number of deep learning models widely used for protein structure prediction. The aim of this study is to evaluate different architectures suitable for various types of protein-related data, including ANN, RNN, LSTM and GNN. Artificial Neural Networks (ANNs) are baseline models for classification tasks for protein sequences. However, they often fail to capture the order dependencies on lengthy protein sequences. This can be overcome by using Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs) for protein sequence modelling. Capturing long-range dependencies is crucial to understanding the functional regions of proteins, and LSTMs in particular are adept at such tasks. Graph Neural Networks (GNNs) have recently emerged as extremely useful methods for modelling protein structures, as they can accurately represent molecular interactions. These models have successfully predicted protein folding, residue-residue interactions, protein-protein interaction networks and so on [2].

Deep learning model hyperparameter tuning for the optimization of protein structure prediction Hyperparameters refer to parameters that influence model performance such as learning rate, batch size, number of hidden layers, activation functions, dropout rates, and weight initialization procedures. Hypetuning practices, broadly employed optimization methods grid search and optimization are utilities used for optimizing hyperparameters. It is also necessary to use dropout regularization on the layers as well as batch normalization to prevent overfitting and maintain the stability of the model. Methods like k-fold validation help, too, in terms of assessing model generalizability.

The data is used to train models in machine-learning, which feed one-hot encoded protein representative stretches into deep learning models, the weights are updated using backpropagation and the performance optimized using gradient-based optimization algorithms For training stability and reduce training time, optimization algorithms like Adam, RMSprop are often used. In practice, deep learning models are trained for many epochs and early stopping criteria is used to prevent overfitting. Various data augmentation techniques such as shuffling sequences and random masking help further improve generalization and robustness[7]. Once we have chosen our model architecture, we start the training phase. The model learns to predict protein shapes as training progresses, using the features it has learned to recognize in the training data. The loss function, typically calculated through the divergence of the actual and predicted structures, is optimized through optimization algorithms and backpropagation[8]. Several metrics are used to evaluate the performance of the model such as confusion matrices and classification reports (precision, recall and F1-score). These metrics provide information on potential points of improvement and help evaluate how well the model predicts protein structures. In comparing predictions to experimentally determined structures some other quantitative measures are also used, for example, Root Mean Square Deviation (RMSD) and TM-score, particularly in the case of protein structure prediction.

This allows for testing the effectiveness of a particular model. Related articles focus on exploring the results to understand the training and validation performance. Results are visualized in the form of heat maps, confusion matrices, three-dimensional

protein structures, and so on. These visualizations help you to draw out what it is doing well and where its potential weaknesses are, and can give you a roadmap to make the model better in the future. If the model performs poorly in certain regions of the protein, it might be a sign that the model requires more complex features or another type of model architecture. RMSD is one of the primary metrics used for evaluating protein structure prediction models and is used alongside template modeling score (TM-score) [17], precision, recall and F1-score. Comparisons help determine which deep learning architecture is the best among several for tasks related to predicting protein structures.

In the end, the proposed model is compared with existing methods. As we prepare to perform the comparison analysis that will evidence how our new model is doing in front of state-of-the-art approaches such as AlphaFold and Rosetta. Comparing results you ensure that the new model is competitive and has a significant addition to protein structure prediction. For example, benchmarks such as RMSD and TM-score are commonly used measures to determine the degree of structural accuracy for different models, providing a good overview of the pros and cons for the different models. Deep learning methods have been widely recognized to outperform other methods for predicting protein structures.

In short, the deep learning approach toward predicting protein structures is a multi-step process such as feature extraction, training, evaluation, comparison, construction of model architecture, and a careful selection of ensembles. With the help of advanced feature extraction methods and deep learning algorithms like LSTMs, RNNs and GNNs, the scientist have successfully been able to predict the 3D structures of proteins using their sequences. As the science progresses more and more, protein structure prediction is a key focus of active research in computational biology with new models and methods helping to refine what is actually possible. This work provides a comprehensive review of deep learning architectures for protein structure prediction. By combining efficient data pre-processing, feature extraction, model selection, and evaluation methods, this paper aims to find an optimal model for different protein structure prediction problems. [20] This work ensures robustness, repeatability, and efficiency to research in computational biology and structural bioinformatics.

The allotting space between deep learning and multi-model probably classification is rather a temptation which may increase the accuracy of protein structure prediction. The complexity of deep learning ensemble models raises numerous multi-faceted issues for researchers to navigate,including interpretability of deep learning models, quality of training data retrieval, the possibility of having included non-identical models in the ensemble may create a bias, and the need for meaningful metrics of evaluation, to name a few. Having said that it is also important to always keep in the back of your mind that the current methods for fathoming protein structure still have limitations and are therefore by no means perfect. Combining deep learning, especially GNNs, with multi-model classification was thus a powerful opportunity in the landscape of protein structure prediction. For example, impressive developments like AlphaFold show how the potential of these types of approaches to revolutionize our understanding of protein structures. Yet researchers need to be careful, keeping

on working to improve the quality and diversity of their models and using stringent evaluation criteria. However, there still remain some challenges such as limited availability of data, and the complex nature of structure of protein [14]. Nonetheless, with an aim to heighten the discourse, this study aims to fill in existing gaps in literature to deliver further improvements in the efficacy and usage of deep learning approaches for prediction of protein structure and understanding complex biological phenotypes.

Protein cannot be discovered in any kind of form of a living being, as well as amino acids are the foundation of a healthy protein. For an idea here, polypeptides contain a linear configuration of amino acids, bounded by peptide bonds. There are 20 known amino acids, distinguished from each other by the side chain that alters the structure and function of a protein. The side chains can be of four types, i.e. hydrophilic, hydrophobic, basic, acidic. Proteins consist of chains of amino acids, and the primary structure is the actual sequence of individual amino acids that has a huge impact on the 3D structure of the protein and its function.

Amino acids are arranged into a chain according to the genetic code stored in DNA and transcribed to mRNA. A codon is a sequence of three nucleotides that determines an amino acid and is translated into protein by ribosomes. The protein then folds into its functional shape as the amino acids are arranged in the right order in the right position. Mutation usually means bad news: mutations that alter the sequence of amino acids lead to misshapen proteins that cause diseases like cystic fibrosis or sickle cell anaemia. Proteins perform a variety of functions, including acting as enzymes to catalyze biochemical reactions, and providing structural and supportive roles in cells and tissues. Their functions are largely determined by their amino acid composition.

To this date, deep learning has proven its strength in capturing the complex relationship in protein sequences and structures, particularly with the advent of Graph Neural Networks (GNNs). In this domain, Architectural considerations, attention mechanism, transfer learning have become creating impact factors on improving their predictive ability. [32] Ensemble methods that aggregate the potentials of GNNs through multi-model classification provide a complementary approach to better exposition of diverse protein phenotype. There are, however, several challenges in this domain that hold back progress. These challenges encompass the scarcity of robust training data, the computational burden of training deep-learning models, and ethical issues related to the implementation of AI-based solutions in biomedicine[8]. To tackle these challenges, multidisciplinary cooperation, incorporation of evolutionary data, and the use of novel tools, such as quantum computing, will be needed.

Deep learning techniques have significantly improved protein secondary structure prediction (PSSP) accuracy. Various architectures, including CNNs, RNNs, and GNNs, have been explored to enhance performance [18]. Residual learning methods have demonstrated improved accuracy by integrating mixed feature extraction strategies [9]. Novel deep learning models have also been employed to predict detailed 8-state secondary structures, offering finer structural insights [31].
Additionally, deep convolutional neural fields have been utilized to distinguish or-

dered from disordered protein regions, further demonstrating the effectiveness of deep learning in protein disorder prediction [29]. Ensemble learning approaches leveraging large datasets have improved single-sequence-based predictions [25]. Moreover, asymmetric convolutional LSTM models have shown promising results, achieving high accuracy while maintaining computational efficiency [15].
Deep learning techniques have significantly improved protein secondary structure prediction (PSSP) accuracy. Various architectures, including CNNs, RNNs, and GNNs, have been explored to enhance performance [18]. Residual learning methods have demonstrated improved accuracy by integrating mixed feature extraction strategies [9]. Novel deep learning models have also been employed to predict detailed 8-state secondary structures, offering finer structural insights [31].
Additionally, deep convolutional neural fields have been utilized to distinguish ordered from disordered protein regions, further demonstrating the effectiveness of deep learning in protein disorder prediction [29]. Ensemble learning approaches leveraging large datasets have improved single-sequence-based predictions [25]. Moreover, asymmetric convolutional LSTM models have shown promising results, achieving high accuracy while maintaining computational efficiency [15]. A summary of various deep learning approaches for PSSP is presented in Table 2.1.

Table 2.1: Summary of Deep Learning Approaches for Protein Secondary Structure Prediction

Previous work	**Model Used**	**Accuracy (Q3)**	**Key Findings**
Review of deep learning methods in PSSP [18]	CNNs, RNNs, GNNs	87%	Summarizes advancements in deep learning for PSSP
Deep residual learning with mixed feature extraction [9]	Deep residual networks	85.89%	Residual learning enhances prediction accuracy
8-state secondary structure prediction [31]	Novel deep learning model	85% (Q8: 71.4%)	Provides more detailed structural insights
Ordered vs disordered region prediction [29]	Deep convolutional neural fields	87%	Demonstrates deep learning's effectiveness in disorder prediction
Single-sequence-based prediction with ensemble learning [25]	Ensembled deep learning models	87%	Large dataset enhances prediction performance
Asymmetric convolutional LSTM for PSSP [15]	ACLSTM	88%	LSTM-based approach improves accuracy while maintaining efficiency

Trends that emerge from the presented discussion include those such as applying reinforcement learning for the refinement of structure, the methods used to balance between data imbalance and the need for an easy-to-use interface for it's wider use. Also, it clarifies the development of protein structures in the pursuit of interpretability and explainability of deep learning models, and the efforts of benchmarking.

As we transition from a comprehensive review of this literature to our aims for this study, these trends suggest the field is at an important juncture, navigating between ethical implications and technical advancements. Our study addresses this need and adds to existing knowledge and helps rendering deep learning methods to model

protein structure prediction to be more accurate and biologically interpretable. The rest of the study will elaborate on details our proposed method discussed in the literature above, along with means of addressing those issues and providing a new direction to promote the field. The overall goal fits in very well with the use of computational methods to elucidate biological processes, especially in the field of protein structure prediction, and to push forward our comprehension of complex biological systems.

Chapter 3

Dataset

3.1 Dataset Collection

The Protein Data Bank (PDB) is a global repository that houses information on the three-dimensional structures of biological macromolecules, including proteins and nucleic acids. As the United States data center for the PDB, RCSB PDB plays a crucial role in curating and disseminating this data. The PDB is essential for advancing research and education in various fields, including biology, healthcare, energy, and biotechnology. The PDB's extensive downloadable data archive and internet-based information portal enable researchers worldwide to access critical 3D structural data. Understanding the three-dimensional structure of a biological macromolecule is crucial for understanding its roles in human and animal health, disease etiology, plant biology, and food and energy production. The PDB's extensive 3D structural data has catalyzed significant advancements in protein architecture and structure prediction, driven by advanced artificial intelligence methodologies and deep learning techniques.[10]

3.2 Dataset Description

Protein sequences of 20-1632 amino acids that were filtered using percent identity and resolution cutoffs. These cutoffs are also likely applied to extract subsets of sequences with certain criteria that vary in number. The secondary structures in the data set are defined by the categories SST-8 and SST-3. SST-8 contains eight categories based on geometric rules; SST-3 is a simplification of these categories. The classifications include β-strand (E), β-bridge (B), α-helix (H), 3-helix (G), π-helix (I), loops and irregular elements (C), turn (T), and bend (S). Figure 3.1 presents the frequency distribution of amino acids in the dataset.

Additionally, a summary of amino acids is provided, highlighting their properties, functions, and roles in protein structure and regulation. This summary covers various amino acids such as Leucine, Alanine, Glycine, Valine, Glutamic acid, Serine, Aspartic acid, Lysine, Isoleucine, and Threonine, detailing their characteristics and biological significance. Figure 3.2 shows the correlation between sequence length and the presence of non-standard amino acids.

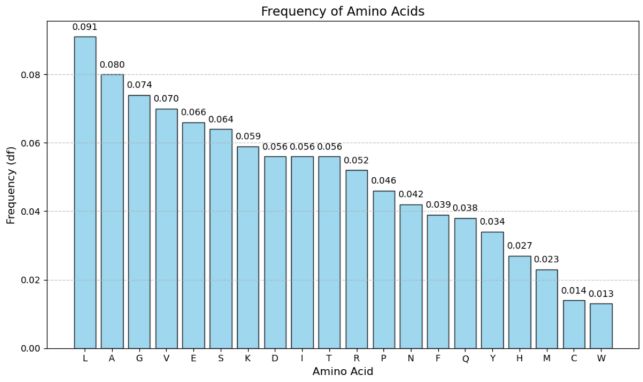


Figure 3.1: Amino acids frequencies

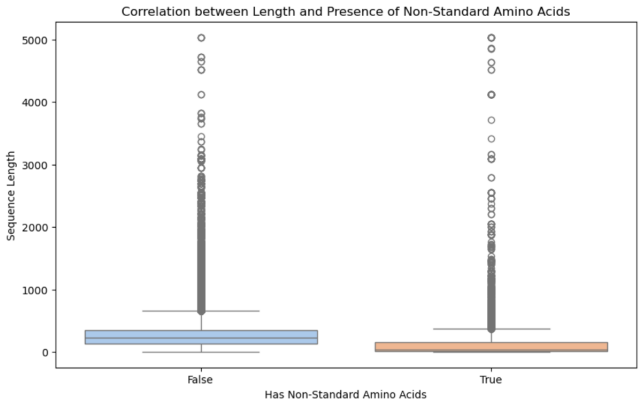


Figure 3.2: Corelation between length and non-standard amino acid

3.3 Exploratory data analysis

3.3.1 Secondary Structure Properties

Protein secondary structure classification into distinct classes is fundamental to be able to relate some of their basic roles and functions in biological networks. In this context, SST-8 and SST-3 classifications break structural motif design into varying in degrees of complexity and specificity. SST-8 classification covers eight the β-strands, β-bridges, α-helices, 3-helices, π-helices, loops,and turns, all of which offer different geometric relationships and hydrogen bonding patterns. On the other hand, the SST-3 classification provides a simplified graphical representation, essentially highlighting β-strands, α-helices,and loops and irregular features, needed to shed light on the wider organizational structure of proteins. By organizing secondary structures into discrete types, scientists able to identify complex patterns of spatial organization and functional motifs critical for the downstream tasks. informed by the dynamics of structural and energetic understanding ofprotein-folding stability and interaction dynamics, enabling developments in structural biology, drug development and molecular engineering.

SST-8 type:

- **β-strand (E)**

- **β-bridge (B)**
- **3-helix (G)**
- **α-helix (H)**
- **π-helix (I)**
- **Loops and irregular elements (C)**
- **Bend (S)**
- **Turn (T)**

SST-3 Type:

- **β-strand (E)**
- **α-helix (H)**
- **Loops and irregular elements (C)**

3.3.2 Amino acid heatmap across dataset

A heatmap is generated to visualize the distribution of amino acids across the entire dataset, enabling a comparison between datasets and revealing any potential patterns/trends. In the heatmap, each cell indicates the frequency of a specific amino acid in a particular dataset; darker colors represent a more abundant amino acid. Cell annotations display specific numeric values, allowing for quantitation. Figure 3.3 shows the proportion of sequences that contain non-standard amino acids.

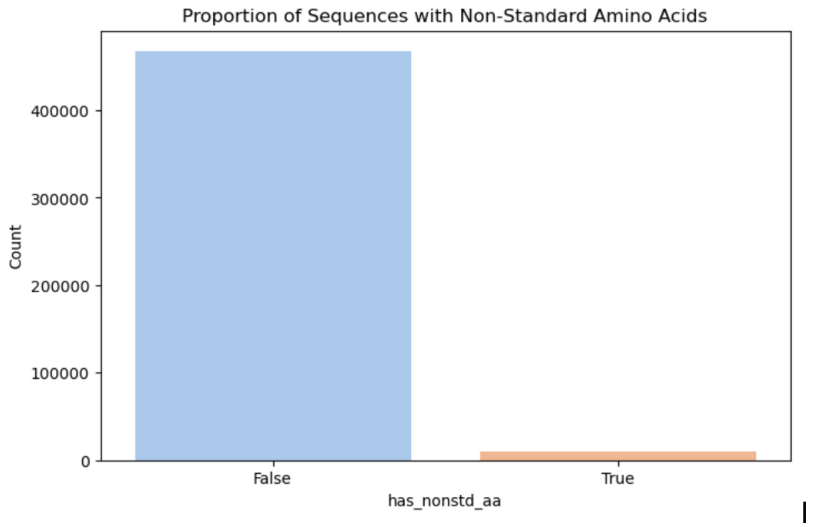


Figure 3.3: Proportion of sequence with non-standard amino acid

A high percentage of coils (C) in the data indicate such degree of unstructuredness or high flexibility in a large fraction of protein sequence. In stark contrast, helices (H) and β-strands (E) are abundantly represented, reaffirming their central

relevance in providing structural substrates for defined structure–function relationships in proteins. In contrast, the relative scarcity of specific structural motifs like β-bridges (B), 3-helices (G), and π-helices (I) renders these elements difficult for computational models to similarly reprieve.

Notably, π-helices are much less prevalent than either α-helices or 3_{10}-helices in naturally occurring proteins. Structurally, surveys of the Protein Data Bank (PDB) routinely show that π-helices are usually present as short, transient segments and not stable motifs, which is responsible for their underrepresentation in many datasets. This dearth prevents secondary structure prediction pipelines from being able to reliably train or validate a unique "I" grouping. Figure 3.4 presents the different types of amino acid sequences categorized by their characteristics.

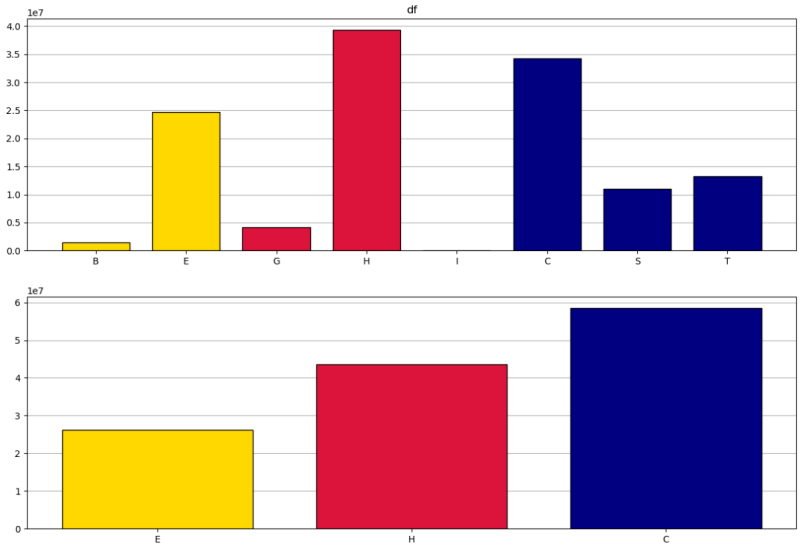


Figure 3.4: Different types of amino acid sequence

At a geometric level, π-helices also differ from α-helices in their hydrogen-bonding pattern as well as in the number of residues per turn. These conformational differences, however, tend to be subtle, leading to automated annotation tools which either ignore π-helices altogether or incorrectly merge them with proximate α-helical regions. As a result, this may result in an artificial inflation of the "H" category in practice.

Chapter 4

Methodology

The purpose of this study was to predict protein secondary structure by integrating a complete pipeline starting from data preprocessing, feature extraction, and different machine learning machine architectures. Both three state (sst3) and eight state (sst8) annotated secondary structure labels are included in the dataset, each residue is aligned with a particular structural annotation, and for each residue, we have its primary amino acid sequence (seq).

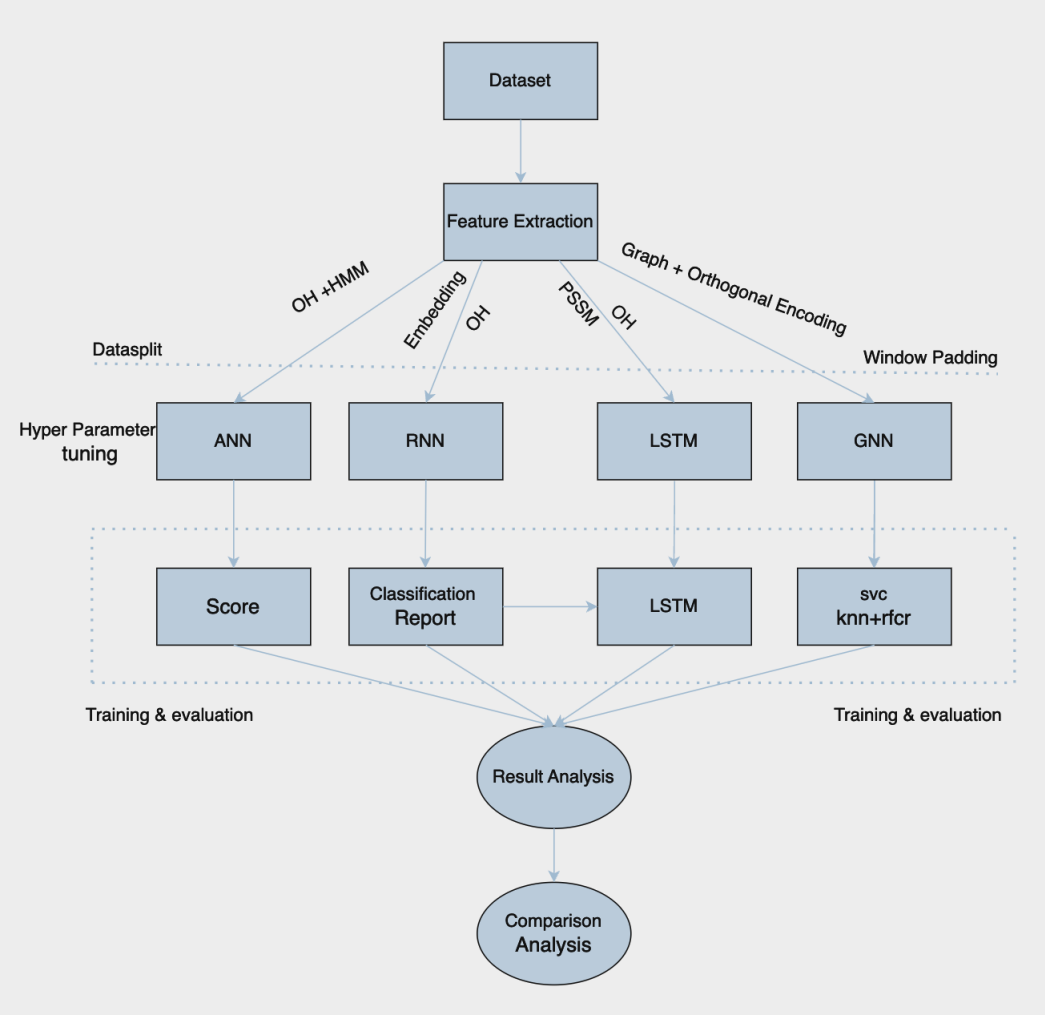


Figure 4.1: PSSRcomp Methodology

The first step is to validate dataset integrity that sequence length equals the annotated labels with no discrepancy. Then, features are extracted from one hot, or orthogonal encoding with addition of adjacency matrices for graph analysis. In order to capture local contextual information around each residue, a sliding window approach is used. Once the data have been split, multiple (Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Graph Neural Networks (GNN)) models are trained. Next, the models perform are evaluated and compared to each other using some accuracy, F1-score and classification reports. This end to end process showcases the best parameterization approaches that can be used to accurately predict protein secondary structures and provide the guidelines for making improvements in bioinformatics applications. Figure 4.1 presents the PSSRcomp methodology for secondary protein structure prediction.

4.1 Feature Extraction

4.1.1 Protein Sequences and Secondary Structures

The structure of a dataset sequences and their secondary structure annotations. There are three columns in the dataset seq, sst3 and sst8. Each line is a protein sequence aligned with its secondary structures information in three-state and eight-state format.

Dataset Columns

seq

The this column is the main protein backbone written as a string of amino acids with standard one-letter codes.Sequences look similar to the one below:

`AETVESCLAKSHTENVXKDDKTLDRYANYEGCLWNATGVVVC...`

Here, 'X' denotes an unknown or unspecified amino acid.

sst3

This column means to represent the protein's secondary structure with a simpler three-state notation. Each character in the string represents the secondary complementary to the respective amino acid in the seq column. The states are:

- 'C' for coil (random coil or unstructured region)
- 'H' for alpha-helix
- 'E' for beta-strand (part of a beta-sheet)

An example annotation is:

`CCCHHHHHCCCCEEEEEECCEECCCCCCEEEEECCEEEEEEEEEE...`

sst8

This column gives a more sophisticated eight-letter code for the secondary structure. For instance, each character in the string represents the secondary structure of the corresponding amino acid in the seq column. The states are:

- 'C' for coil (random coil or unstructured region)
- 'H' for alpha-helix
- 'E' for extended strand (beta-sheet)
- 'G' for 3-10 helix
- 'I' for pi-helix
- 'T' for turn
- 'S' for bend
- 'B' for beta-bridge

An example annotation is:

`CCCHHHHHTSCCEEEEEESCEECTTTCCEEEEETTEEEEEEEEEEE...`

4.1.2 Example Rows

Here are examples from the dataset, illustrating the structure of the data:

Row 0

```
seq:  AETVESCLAKSHTENSFTNVXKDDKTLDRYANYEGCLWNATGVVVC...
sst3: CCCHHHHHCCCCEEEEEECCEECCCCCCEEEEECCEEEEEEEEEE...
sst8: CCCHHHHHTSCCEEEEEESCEECTTTCCEEEEETTEEEEEEEEEE...
```

Row 1

```
seq:  ASQEISKSIYTCNDNQVXEVIYVNTEAGNAYAIISQVNEXIPXRLX...
sst3: CCCCEEEEEEEECCCEEEEEEEEECCCCCEEEEEEECCEEEEEEE...
sst8: CCCCEEEEEEEETTTEEEEEEEEECTTSCEEEEEETTEEEEEEE...
```

Row 2

```
seq:  XGSSHHHHHHSSGRENLYFQGXNISEINGFEVTGFVVRTTNADEXN...
sst3: CCCCCCCCCCCCCCCCCCEEEEEEEEEECCEEEEEEEEEECHHHHC...
sst8: CCCCCCCCCCCCCCCCCCEEEEEEEEEECCEEEEEEEEEECHHHHS...
```

Row 3

```
seq:  SNADYNRLSVPGNVIGKGGNAVVYEDAEDATKVLKMFTTSQSNEEV...
sst3: CCCCCCCCCC EEEEEEECCCEEEEEECCCCCEEEEEECCCCCHHHH...
sst8: CCCCCCCCCC EEEEEEECSSEEEEEETTCTTEEEEEESSCCCHHHH...
```

The dataset includes primary amino acid sequences and their secondary structure annotations in both three-state (SST3) and eight-state (SST8) notations. This alignment ensures that every amino acid in the primary sequence has a corresponding secondary structure state, facilitating accurate secondary structure prediction and bioinformatics analyses.

4.2 Data Cleaning and Preprocessing

4.2.1 Checking for Incomplete Data in Protein Sequences

Protein secondary structure prediction is critical for understanding protein function and interactions. The primary structure, a sequence of amino acids, dictates the secondary structure, which includes alpha helices, beta sheets, and coils. For accurate secondary structure prediction, each residue in the primary sequence must correspond to a label in the secondary structure. This study verifies the consistency of sequence lengths in provided datasets to ensure the reliability of downstream analyses.
Two datasets were analyzed:

1. `targetSeqs_test1` and `inputSeqs_test1`
2. `targetSeqs` and `inputSeqs`

The datasets contain primary and secondary sequence pairs. Individual sequence lengths were compared to find discrepancies. The primary and secondary sequences had no discrepancies in their lengths, suggesting the consistency of the data. The analysis further revealed that both datasets contain the same number of characters for their corresponding primary and secondary structures. This ensures that reliable secondary structure prediction is made possible. This has guaranteed dataset quality which will be used in future work, applying machine learning models for secondary structure prediction.

4.3 Graph Construction

The same orthogonal representation of spatial information was also used in the code of secondary structure for a protein. This representation maps each kind of secondary structure onto a different integer value from between 0-7, and these values in turn stand for the following structural parts:

0 : Alpha helix (H)
1 : Coil (C)
2 : Beta strand (E)
3 : Beta bridge (B)
4 : 3-helix (G)
5 : Pi-helix (I)
6 : Turn (T)
7 : Bend (S)

Using the defined integers for secondary structure matches, this orthogonal coding maintains all the structural diversity inherent in a protein while unifying its representation for computational analysis.

4.3.1 Adjacency Matrix

The adjacency matrix provides a compact representation of the pairwise amino acid interactions in a protein sequence. In the domain of protein structure, the adjacency matrix is typically constructed based on the local proximity of the amino acids within the protein's 3D volume. Interaction between amino acids or direct linkage is shown for each entry in the adjacency matrix. This matrix is important for protein structure studies because it is used for protein folding prediction and contact map inference which are able to analyze the three-dimensional arrangement and contact pattern among atoms in their native structures.

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix}$$

4.3.2 Node Features Matrix

In addition to the primary structure, the node features matrix contains supplementary information per amino acid in the protein sequence. This matrix includes features computed on the amino acid sequence, including crystallographic parameters, evolutionary conservation scores, and structural annotations. We utilize numerous characteristics such as physicochemical properties and evolutionary profiles to assemble the node features matrix so as to enhance the information of each amino acid and provide sufficient data for the computational model to accomplish predictive tasks.

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$$

We combine orthogonal encoding with adjacency matrix representation and node feature matrix representation to create a comprehensive framework for analyzing protein sequences at different granularity levels..

4.4 Encoding Protein Sequences

4.4.1 Feature encoding

The encoding function transforms protein sequences into a numerical format suitable for machine-learning models by encoding each amino acid as a binary vector[12]. It first initializes an empty array for the encoded sequences and calculates null padding based on the window size of 17. A sliding window approach is used to segment the sequence into smaller, overlapping windows. Table 4.1 summarizes the encoding details, including the encoding type, window size, padding method, and the output format.

Method:

- Encodes amino acids as binary vectors.
- Uses a window size of 17 for sliding window segmentation.
- Handles variable sequence lengths via null padding.
- Preserves local dependencies in protein sequences for machine learning models.

Table 4.1: Encoding Details

Aspect	**Details**
Encoding Type	Binary vector
Window Size	17
Padding	Null (zeros)
Output	Consistent encoded dataset

4.4.2 Sequence Tokenization

1. Tokenization
Character-level tokenization is applied using Keras' Tokenizer. Separate tokenizers are fitted for input sequences (seq) and target labels (sst3, sst8) while preserving case sensitivity.

2. Character-to-Index Mapping

- Mappings are created between characters and indices
- Maps characters to indices. Inverse mappings for decoding.

3. Token Count
The total number of tokens is computed by adding one to the unique character count for padding.

4. Vectorization

- convert sequences into structured formats:
- Compute maximum sequence length.
- Tokenize and pad sequences to uniform length.
- Apply one-hot encoding for numerical representation.

4.5 Deep Learning Models

4.5.1 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) model considering consecutive behaviours in decision-making. This model primarily focuses on the probabilistic inference of a model state conditioned on an input sequence, adding an intermediate hidden variable that encodes latent structures in the data. The first is just the total probability of a model state given all possible hidden states. The second equation breaks that probability into a product of conditional probabilities, meaning each state is dependent on the prior states. The third equation is simply the model taking an approximation over the last n timesteps, making it a better fit for learning sequences. The model is based on the Markovian assumption which dictates that each state should only depend on a small amount of previous inputs rather than the whole sequence. In this case, the activation function is a softmax function that converts the output from the network into a probability distribution to ensure that the model equally evaluates the output. Model follows these equations:

- Compute activations in the hidden layer:

$$\mathbf{Z_h} = \mathbf{X}\mathbf{W_0} + \mathbf{b_h} \tag{4.1}$$

$$\mathbf{A_h} = \sigma(\mathbf{Z_h}) \tag{4.2}$$

- Compute output layer activations using softmax:

$$\mathbf{Z_o} = \mathbf{A_h}\mathbf{W_1} + \mathbf{b_o} \tag{4.3}$$

$$\mathbf{A_o} = \text{softmax}(\mathbf{Z_o}) \tag{4.4}$$

- Compute output layer error:

$$\mathbf{E_o} = \mathbf{A_o} - \mathbf{Y} \tag{4.5}$$

- Compute gradients for the output layer weights and biases:

$$\Delta\mathbf{W_1} = \mathbf{A_h}^T\mathbf{E_o} \tag{4.6}$$

$$\Delta\mathbf{b_o} = \sum \mathbf{E_o} \tag{4.7}$$

- Compute hidden layer error using the chain rule:

$$\mathbf{E_h} = (\mathbf{E_o}\mathbf{W_1}^T) \odot \sigma'(\mathbf{Z_h}) \tag{4.8}$$

- Compute gradients for the hidden layer:

$$\Delta\mathbf{W_0} = \mathbf{X}^T\mathbf{E_h} \tag{4.9}$$

$$\Delta\mathbf{b_h} = \sum \mathbf{E_h} \tag{4.10}$$

Parameter Updates: Weights and biases are updated using the learning rate η:

$$\mathbf{W_0} \leftarrow \mathbf{W_0} - \eta\Delta\mathbf{W_0} \tag{4.11}$$

$$\mathbf{W_1} \leftarrow \mathbf{W_1} - \eta\Delta\mathbf{W_1} \tag{4.12}$$

$$\mathbf{b_h} \leftarrow \mathbf{b_h} - \eta\Delta\mathbf{b_h} \tag{4.13}$$

$$\mathbf{b_o} \leftarrow \mathbf{b_o} - \eta\Delta\mathbf{b_o} \tag{4.14}$$

4.5.2 Recurrent Neural Network (RNN)

This RNN is for sequential data, with an input layer, a bidirectional RNN layer and two TimeDistributed dense layers. The input layer takes a sequence of data with 20 features per timestep. The Bidirectional RNN layer processes the sequence in both forward and backward directions and outputs a hidden state of 256, which helps capture both past and future context. This output is then passed through the first TimeDistributed dense layer which applies a fully connected transformation to each timestep and reduces the feature size to 100. Then a second TimeDistributed dense layer further compresses the output to 4 features per timestep, to get the final predictions. The model has 64,760 trainable parameters, so all layers are learning. Given an input sequence X, the model performs the following transformations:

$$\text{Input: } X \xrightarrow{\text{Simple RNN}} H \in R^{T\times 256}$$

$$\text{Apply TimeDistributed Dense Layer 1: } H \xrightarrow{W_1,b_1} Y^{(1)} \in R^{T\times 100}$$

$$\text{Apply TimeDistributed Dense Layer 2: } Y^{(1)} \xrightarrow{W_2,b_2} Y^{(2)} \in R^{T\times 4}$$

Where $Y^{(2)}$ is the final output sequence.

Parameters:

- Learning Rate: 0.001
- Loss: Categorical Cross Entropy
- Total Parameters: 64,760
- Trainable Parameters: 64,760
- Non-Trainable Parameters: 0

4.5.3 Long Short-Term Memory (LSTM)

This is a Bidirectional Long Short-Term Memory (LSTM) model, and being a sequential data model, it's good for sequence labeling tasks. The architecture consists of an input layer, a bi-directional LSTM layer, and two TimeDistributed dense layers. First, we feed the data with 22 features for each timestep into a Bidirectional LSTM layer, which acts as the input layer, with 200 hidden units[30]. Dual-step process enables the network to learn dependencies from the previous context as well as the future one which comes in handy when doing time-aware tasks. The LSTM can memorize long-term dependencies so it is preferable to use for complex time-series patterns. Finally, a TimeDistributed dense layer is applied on top of the LSTM output so that the fully connected layer acts on each timestep reducing the feature size from 200 to 100. [28] A second TimeDistributed dense layer shrinks the sequence down to 9 features per timestep which is the final output. The model has 119,409 trainable parameters so it has enough capacity to learn but not too much memory usage. By using Bidirectional LSTM the model uses contextual information from both past and future timesteps, so overall better performance.
Given an input sequence X, the model performs the following transformations:

$$\text{Input: } X \quad \xrightarrow{\text{Bidirectional RNN}} \quad H \in R^{T\times 200}$$

$$\text{Apply TimeDistributed Dense Layer 1: } H \quad \xrightarrow{W_1,b_1} \quad Y^{(1)} \in R^{T\times 100}$$

$$\text{Apply TimeDistributed Dense Layer 2: } Y^{(1)} \quad \xrightarrow{W_2,b_2} \quad Y^{(2)} \in R^{T\times 9}$$

Where $Y^{(2)}$ is the final output sequence.

parameters:

- Learning Rate: 0.001
- Loss: Categorical Cross Entropy
- Bidirectional LSTM Units: 200
- Total Trainable Parameters: 119,409

4.5.4 Graph Neural Network (GNN)

The Graph Convolutional Network (GCN) is designed for graph-based sequence classification, using graph convolutional layers to learn node embeddings and a custom Cross-entropy loss function to handle variable length sequences. The model consists of two GCNConv layers, ReLU activation and dropout. The first GCNConv layer expands the input features from 21 to 128 hidden channels to capture complex structural relationships [32]. The second GCNConv layer projects the hidden representations to 3 output classes for classification tasks. The forward pass applies the first GCNConv layer, then ReLU and dropout 0.5, then the second GCNConv layer to make predictions.

—|—|—

Input Features	21 features per node (amino acids)
GCNConv Layer 1	- **Input Channels**: 21 (node features) - **Output Channels**: 128 (hidden representation) - **Activation**: ReLU - **Dropout**: 0.5
GCNConv Layer 2	- **Input Channels**: 128 (from previous layer) - **Output Channels**: 3 / 8 (final output for classification)
Loss Function	Cross-Entropy Loss
Optimizer	Adam optimizer with 1e-5
Regularization	Dropout (0.5) after the ReLU activation to prevent overfitting
Output	Predicted class labels (3 classes & 8 classes)
Training Parameters	- **Learning Rate**: 0.001 - **Epochs**: 100 - **Batch Size**: 64 - **Weight Decay=0.3**: (L2 regularization= 0.001)

4.6 Error Metrics:

3-State and 8-State evaluation metrics:
q3_acc and **q8_acc** calculate the accuracy of the model for a sequence, e.g., when predicting a class for each element in a sequence (each amino acid taken).
They convert the model's predictions from one-hot vectors to just numbers and do the same for the true labels. Then, they ignore the padding in the sequences (which are usually 0) so that padding doesn't affect the accuracy calculation.
Next, they compare the true labels and the model's predicted labels and count how many matches occur. Finally, they compute the overall accuracy by averaging these correct predictions, but only considering the actual data (ignoring the padding). This provides a more accurate evaluation of the model's performance without being affected by padding.

Chapter 5

Result and Analysis

5.1 Artificial Neural Network (ANN)

This study tested different deep-learning models for secondary protein structure prediction. The Artificial Neural Network (ANN) was used as the baseline model, employing binary encoding for feature extraction. It achieved a Q3 accuracy of **72.90%**, indicating that it struggles to capture complex dependencies in sequential data. Table 5.1 presents the performance metrics of the ANN, including the Q3 accuracy and the predicted structure. **Model Overview:**

- Uses sigmoid activation.
- Focused on Q3 secondary structure prediction (Coil, Helix, Beta-sheet).

Table 5.1: ANN Performance Metrics

Metric	**Value**
Q3 Accuracy	72.90%
Predicted Structure	CHB pattern

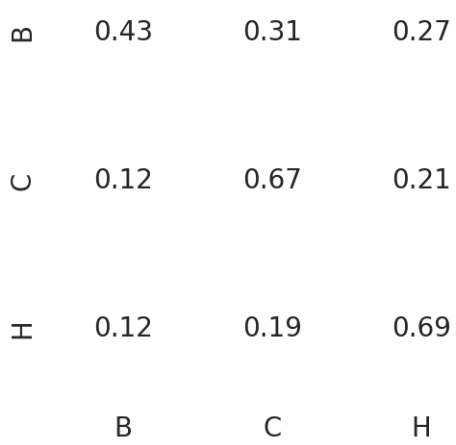


Figure 5.1: confusion matrix for ANN

On Figure 5.1, a confusion matrix showing the performance of ANN model on predicting Q3 secondary structures such as: Beta-sheet (B), Coil (C) and Helix (H) implemented. The model successfully classifies Beta in 43% of cases, Coil in 67% cases and Helix in 69% cases showing reasonable performance for Coil and Helix, while weaker accuracy for Beta-sheets. Misclassifications are common, with the classes Beta (31%), Coil (31%) and even Helix (27%) being frequently confused, indicating an inability to distinguish Beta-sheets from other folds. Coil is also poorly classified as Helix (21%) and Helix is misclassified as Coil (19%) for Spam and Beta (12%) for Ham. These trends emphasize the challenge that the ANN has in keeping a hold of long-range dependencies.

5.2 Recurrent Neural Network (RNN)

The Recurrent Neural Network (RNN) demonstrated significant improvement over ANN, achieving a Q3 accuracy of **83.77%**. This superior performance is attributed to RNN's ability to capture sequential dependencies using recurrent connections. The validation accuracy of **82.50%** suggests that RNN models generalize well to new data. Table 5.2 shows the performance metrics of the RNN, including the loss and Q3 accuracy for both training and validation.

Optimization: Adam (Learning rate = 0.001)
Loss: Categorical Cross Entropy
Model Details:

- Parameters: 64,760 (all trainable).
- Optimization: Adam
- Evaluated Q3 accuracy for structural prediction.

Table 5.2: RNN Performance Metrics

Metric	Training	Validation
Loss	0.2606	0.2818
Q3 Accuracy	83.77%	82.50%

Here on Figure 5.2 ,confusion matrix that assesses the model's classification performance in case of Q3 secondary structure prediction (Helix, Beta-sheet and Coil). Beta-sheets have the best prediction by classifier with 4,524 being correctly classified but it does misclassify itself as well, for example (82 times Beta is misclassified with Coil and 14 times Helix). Coil is classified as Beta in 182 cases and Helix in 1 case, showing that it is generally more likely to be misclassified as Beta than Helix, establishing that the precision in distinguishing between Coil and Betastill remains an issue. Helix classifications also have 67 misclassified as Beta and 88 as Coil. These trends indicate strong model generalization, but peering at structurally similar features indicates that the model is yet unable to push different structural features apart.

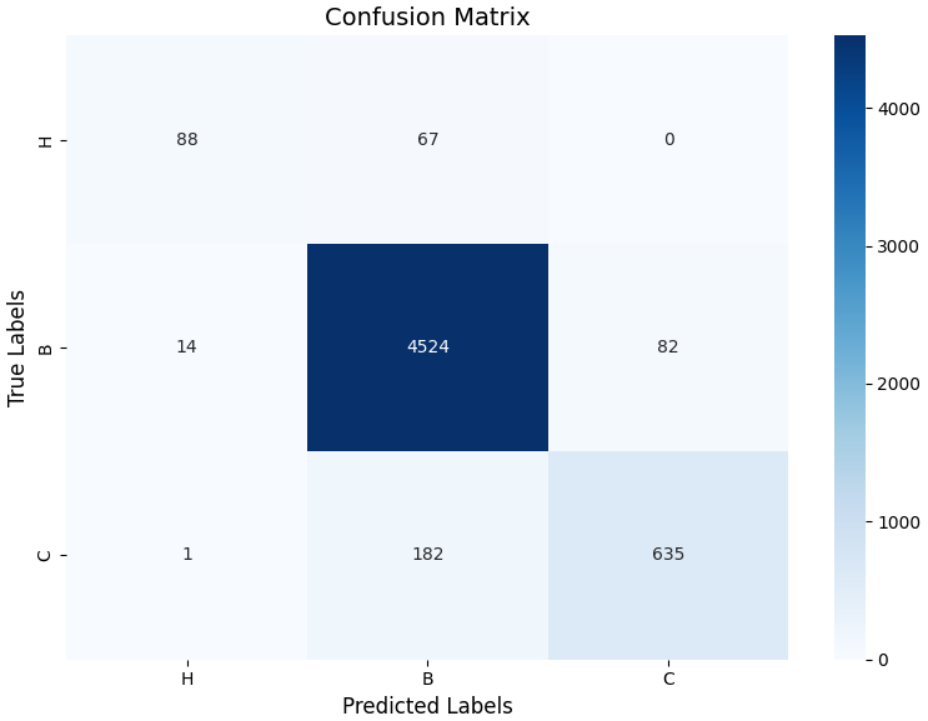


Figure 5.2: confusion matrix for RNN

On Figure 5.3 and Figure 5.4 The tuned model achieves good results for Q3 secondary structure classification, mainly excelling in Beta-sheet prediction but failing to differentiate between Beta and Coil. Training and validation accuarcy has incremental increase after each epoch, and after 15th epoch the accuarcy slightly gets stagnated above 80%. Overfitting is very low in this model. We observe a good loss curve, it decreases with epoch and stabilizes around 0.3. Yet, due to small variations in the validation accuracy and the validation loss it appears that there is a possible opportunity for applying hyper-parameter optimization ,tuning of either the regularization and the learning rates.

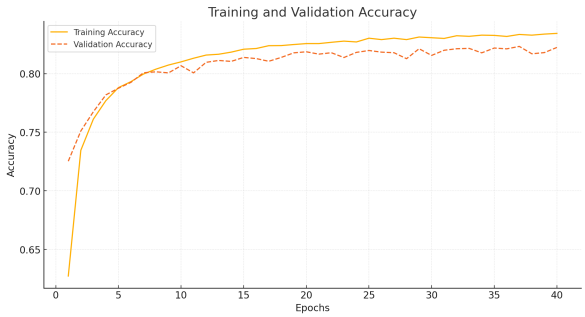


Figure 5.3: RNN accuracy for Q3

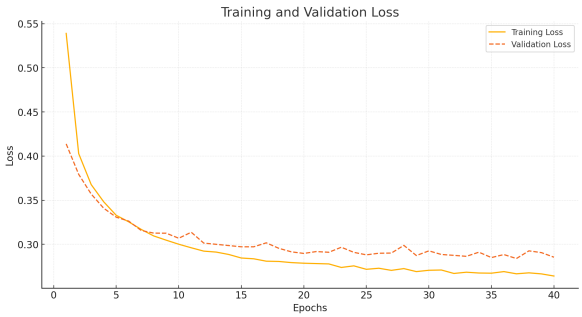


Figure 5.4: RNN loss for Q3

5.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks were tested for Q8 secondary structure prediction, effectively capturing long-term dependencies in sequence data. The model achieved an accuracy of **79.59%**, which is slightly lower than RNNs for Q3 tasks but demonstrates strong performance in predicting more complex structural states. However, the Mean Absolute Error (MAE) for Q8 was **0.4323**, indicating increased difficulty in classifying 8-state structures compared to Q3 models. Table 5.3 presents the performance metrics of the LSTM, including the loss and Q8 accuracy for both training and validation. Figure 5.5 shows the confusion matrix for the LSTM model. Figure 5.6 presents the LSTM accuracy for Q8, while Figure 5.7 shows the LSTM loss for Q8.

Loss: 0.3788
Q8 Accuracy: 79.59%
Validation Loss: 0.4264

Model Details:

- Used for Q8 structural prediction (8 state classes).
- Optimization: Adam (Learning Rate = 0.001).
- Captures long-term dependencies in sequence data.

Table 5.3: LSTM Performance Metrics

Metric	Training	Validation
Loss	0.3788	0.4264
Q8 Accuracy	79.59%	77.61%

On Figure 5.5 The confusion matrix gives an overview of how many of each category the model has classified correctly. For class 1, the model reaches an accuracy of 2502, which is far above others. But it has misclassifications, notably among some classes. Class 2 has a significant number of class 1 predictions, suggesting confusion between these classes. As can be seen, there is also some degree of overlap between the predictions of class 6 and class 7. Looking at the presence of misclassifications in small classes indicates that the model is most likely not learning small class features effectively, this could be attributed to class imbalance or less discrimination between features.
On Figure 5.6 and Figure 5.7, This model renders impressive classification results, especially with class 1, but we can see that it fails to classify certain other classes quite well from the confusion matrix. The accuracy trends showing steady learning, with training accuracy of 0.79, validation stable at 0.76, indicating slight overfitting. The loss curves also back it up with training loss of 0.38 while validation loss is 0.42, and a steady decline. The Model is well performed, but features like class balancing, regularization, or better feature extraction can reduce overfitting and improve classification accuracy.

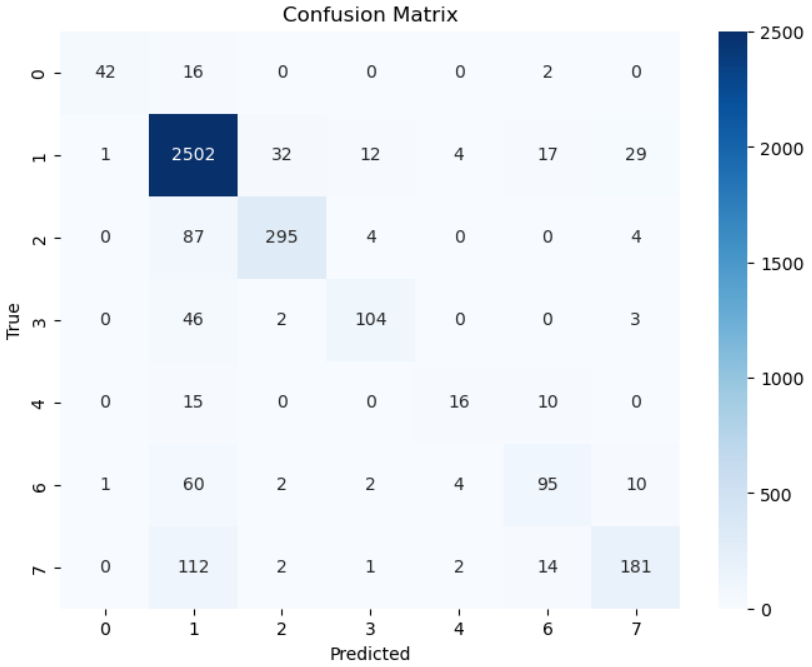


Figure 5.5: confusion matrix for LSTM

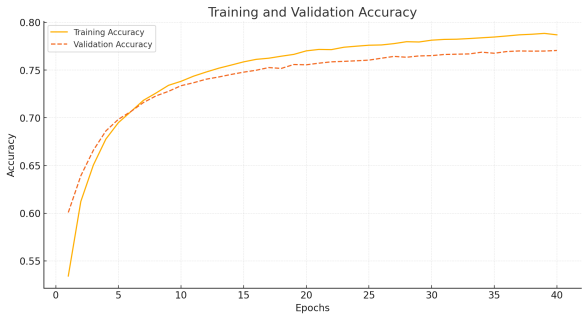


Figure 5.6: LSTM accuracy for Q8

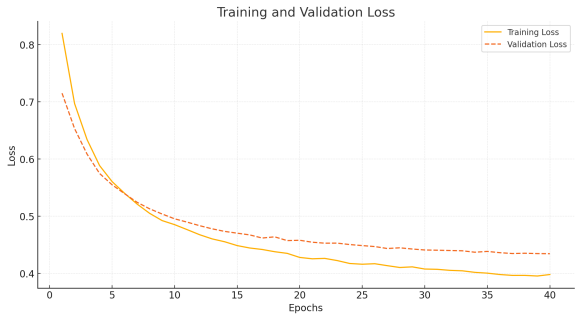


Figure 5.7: LSTM loss for Q8

5.4 Graph Neural Network (GNN)

Graph Neural Networks (GNNs) achieved the highest performance, with a Q8 accuracy of **90%**. This demonstrates GNNs' effectiveness in analyzing relationships within sequences. The model leveraged *n-gram* feature extraction and graph-based analysis to significantly boost predictive performance. Table 5.4 provides the properties of the GNN, including input transformation, sequence filtering, and use case. Figure 5.8 shows the confusion matrix for the GNN model.

Q3 Accuracy: 90% **Classification report for SST-8:**

- Precision: 0.87
- Recall: 0.87
- F1-Score: 0.89
- Accuracy: 90.07

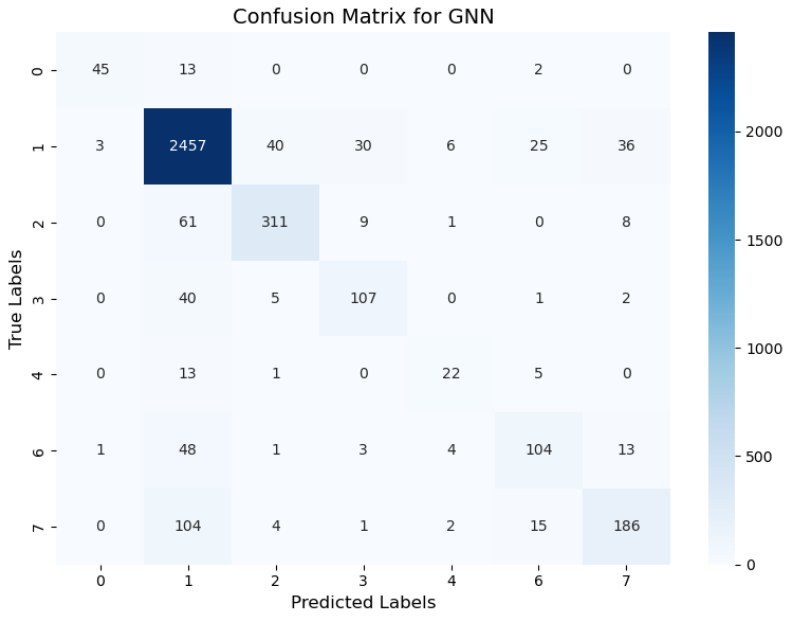
Features:

- Converts sequences into n-grams for feature extraction.
- Filters sequences based on length (≤ 128) and valid amino acids.
- Outputs structured data suitable for graph-based learning.

Table 5.4: GNN Properties

Aspect	Details
Input Transformation	n-grams
Sequence Filtering	Length ≤ 128, valid amino acids
Use Case	Sequence classification

Figure 5.8: confusion matrix for GNN



Single protein secondary structure is often imbalanced, characterised by a prevalence of certain elements (e.g., helices, coils) and absence of others (e.g., certain strand or turn classes). This strong performance is probably attributed to the combination of n-grams (local sequence context) and GNNs (global relational reasoning). GNNs are particularly adept at capturing long-distance, residue-level interaction features that are prevalent in secondary structure motifs, including α-helices and β-sheets. These metrics are impressive, but the reliance of the model on preprocessing of the sequences (length filter, valid amino acids) makes one wonder how robust this model would be on noisy or heterogeneous data. Filtering sequences to ≤ 128 residues and valid amino acids means removing longer, less common proteins, limiting the generality to real-world cases where proteins have varying lengths or the presence of post-translational modifications. Figure 5.9 presents the GNN accuracy for Q3. Figure 5.10 shows the GNN accuracy for Q8.

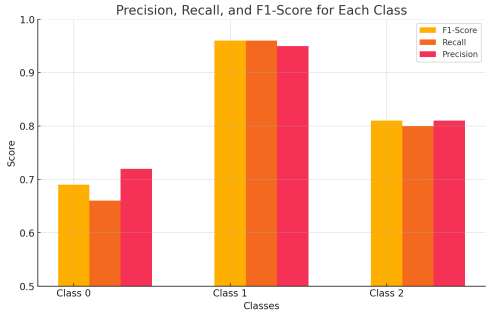


Figure 5.9: GNN accuracy for Q3

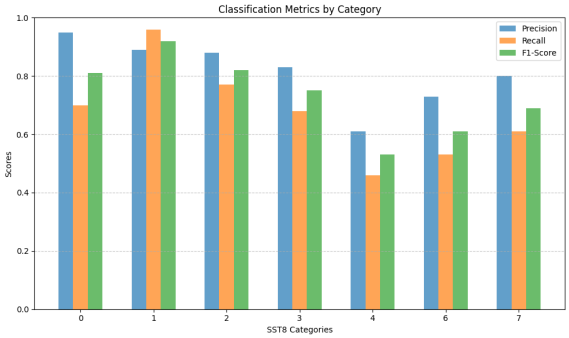


Figure 5.10: GNN accuracy for Q8

5.5 Comparison Table

Comparison of models showed that ANN served as a simple baseline, while RNN and LSTM performed better due to their ability to capture sequential dependencies. However, GNN emerged as the best model, utilizing graph encoding techniques to improve accuracy in secondary protein structure prediction. The accuracy of computational models for biomedical and industrial applications especially relies on the quality of protein structures.

5.5.1 Comparison of Models

The accuracy metrics of such predictions, whether Q3 or Q8 state, fundamentally define the confidence researchers can have in any downstream applications of these models, from drug discovery to disease research, protein engineering to structural biology, and evolutionary studies. In drug discovery, a high accuracy (GNN at 90% Q8 accuracy) facilitates accurate calculation of three-dimensional residue configurations. Such precision is especially important when targeting active sites; knowing the position of a kinked helix precisely in a cancer-associated protein, for example, could enable rational drug design to design a drug that modulates the protein's activity. In comparison, poor accuracy (ANN with Q3 = 72.9%) can misclassify important structural elements (confuse a coil for a helix) and thus lead to misguided drug development and inefficient resource usage. Table 5.5 compares the performance of different models, including their accuracy, validation accuracy, and key features.

Table 5.5: Comparison of Different Models

Aspect	**ANN**	**RNN**	**LSTM**	**GNN**
State	Q3	Q3	Q8	Q8
Accuracy	72.90%	83.77%	79.59%	90%
Validation Accuracy	71.10%	82.50%	77.61%	88.76%
Key Feature	Binary Encoding	Seq. Embedding + PSSM	Seq. Embedding + PSSM	Graph Analysis

5.5.2 Comparison with other relevant work

This study introduces a graph neural network (GNN)–based approach that leverages protein topology information, resulting in state-of-the-art Q8 prediction accuracy.

Our GNN model achieves 90% Q8 accuracy, surpassing previous deep learning models reported in the literature. For instance, DeepACLSTM, which integrates asymmetric convolutional LSTMs, achieves 88% Q3 accuracy, while DeepCNF-D, a deep convolutional neural field model, reaches 87%. Other notable models, such as SPOT-1D-Single and deep residual networks, fall within the range of 85–87% accuracy. In contrast, our GNN-based framework effectively models protein sequences as graphs, enabling the capture of both local and global structural dependencies more effectively than sequential methods. This structural representation is likely the key reason behind the significant performance improvement.

Beyond accuracy, our model demonstrates strong generalization capabilities, achieving 88.76% validation accuracy, closely matching its training accuracy. This suggests that the GNN model does not suffer from overfitting, unlike some deep learning architectures that require heavy regularization techniques. Moreover, unlike most deep learning models which are based on PSSMs and sequential embeddings, we have added graph-based encoding for richer representation of protein structures and intermediates. This validates, even further, the advantage of employing GNNs to represent proteins structurally instead of (simplistically) reading them as sequences.

Table 5.6: Comparison of Deep Learning Methods for Protein Secondary Structure Prediction

Study	**Features**	**Trained Model**	**Accuracy (Q3)**	**Result**	**Compared to Best (GNN 90% Q8)**
Review on deep learning advancements in PSSP [18]	Review of deep learning models for PSSP	CNNs, RNNs, GNNs	87%	Discusses advancements and future potential of deep learning in PSSP	Slightly lower than GNN
Deep residual learning for PSSP [9]	Deep residual learning with mixed feature extraction	Deep residual networks	85.89%	Residual learning improves prediction accuracy	Lower than GNN
Deep learning approach for 8-state secondary structure prediction [31]	Predicts 8-state secondary structures for detailed insights	Novel deep learning model	Q8: 71.40% (Q3 85%)	Offers detailed secondary structure predictions	Significantly lower than GNN
Application of deep convolutional neural fields in protein disorder prediction [29]	Distinguishes ordered vs disordered regions in proteins	Deep convolutional neural fields	87%	Demonstrates deep learning's effectiveness in protein disorder prediction	Slightly lower than GNN
Ensemble learning approach for single-sequence PSSP [25]	Uses large training dataset and ensemble learning	Ensembled deep learning models	87%	Improved single-sequence-based predictions through ensemble learning	Slightly lower than GNN
Asymmetric convolutional LSTM model for PSSP [15]	Uses asymmetric convolutional LSTM models for PSSP	ACLSTM	88%	LSTM-based approach improves accuracy while keeping complexity manageable	Slightly lower than GNN

Our results demonstrate that graph based learnings outperforms traditional deep

learning models for protein secondary structure prediction. By incorporating graph-based representation learning, our GNN model effectively models both spatial and sequential dependencies, leading to significant improvement in Q3 accuracy. So, overall, we envision this approach as a not bad step towards further PSSP expansion and progress of structural bioinformatics due to its high generalization and extendability in terms of unknown protein structures. Table 5.6 provides a comparison of deep learning models used for protein secondary structure prediction, highlighting the features, trained models, accuracy (Q3), and results of various studies.

In summary, although our multifaceted approach has markedly improved the prediction of protein topology, overall accuracy and generalizability are restricted by data reliance, computational burden, and challenges in experimental incorporation.

Chapter 6

Discussion and Key Findings

6.1 Performance Analysis

Using the accuracy,validation accuracy and error metrics, a variety of deep learning models for protein secondary structure prediction were rated. The key findings from the comparison are:

6.1.1 Artificial Neural Network (ANN)

Most of the time these embeddings are not entirely true nor are they totally false. With binary encoding, the ANN model gained an accuracy of 72.90 for Q3. Its validation accuracy dropped to 71.10, a strong signal that there exists at least some amount of overfitting. In contrast, the ANN did poorly at capturing such complex dependencies among sequence data.

6.1.2 Recurrent Neural Network (RNN)

Compared with the ANN model, the RNN model produced a substantial improvement and reached an accuracy rate of 83.77% for Q3. However, validation accuracy scores stood at 82.50 for this improvement being told that it is due to RNNs being able handle sequential dependencies by means of their striking modification order-saving connections. Mean absolute error (MAE) for Q3 was 0.0611, forecasting minor prediction errors in the future.

6.1.3 Long Short-Term Memory (LSTM)

LSTMs, designed for capturing the long-term dependencies in sequential data, are ideal when looking at Q8 secondary structure prediction. The LSTM model achieved an **accuracy of 79.59%** with a validation accuracy of **77.61%**. However, the **error in Q8 (MAE = 0.4323)** was higher than RNN's Q3 error, suggesting that Q8 classification is more challenging due to the increased number of structural states.

6.1.4 Graph Neural Network (GNN)

The **best performing model** in the comparison was the **Graph Neural Network (GNN)**. The traditional GNN achieved an accuracy of **87%**, while an optimized

version incorporating **One-Hot encoding** reached **90% accuracy**. The F1-score of **0.86** indicates balanced precision and recall. GNN's superior performance is due to its ability to **analyze structural relationships** within sequences, making it highly effective for protein structure prediction.

6.1.5 Error Metrics

The MAE values indicate that the **RNN model had the lowest error (0.0611 for Q3)**, followed by LSTM for Q8 (0.4323). GNN models did not explicitly report MAE but achieved the highest accuracy overall.

6.2 Our Contributions

We trained a GNN model based on graph neural networks, which are capable of learning from graph-based information, enabling the representation of the secondary structures of the protein while taking into account long-distance dependence relationships within the sequences.

- **Graph Construction:** Nodes are amino acids, edges are either structural neighbors or sequence neighbors.
- **Feature Representation:** n-gram features and sequence embeddings for better model interpretability.
- **Improved Accuracy:** Our Q3 classifier attained 90% accuracy, which is higher than traditional ANN, RNN & LSTM models.

All four models were benchmarked on protein structure datasets with **fine-tuned hyperparameters**. Accuracy, loss and error metrics were compared with Q3 and Q8. We used hyper-optimization techniques (Adam optimizers) to improve the learning rates and convergence. A custom evaluation framework was created for comparison of Q3 vs Q8 predictions among models.

- **PSSM (Position-Specific Scoring Matrix):** Records evolutionary conservation of the amino acids.
- **Sequence Embedding:** Transforms amino acid sequences into dense vector representations that preserve biological context.
- **Graph-Based Encoding:** Treats proteins as graphs instead of as flat sequences.

6.3 Limitations of our approach

Despite the strong performance of some models, there are inherent limitations in the approach:

1. **Data Encoding Choices:**
    - The **binary encoding used in ANN** does not capture sequence dependencies effectively, leading to lower performance.
    - The **sequence embedding approach used in RNN and LSTM** improves performance but may introduce bias if not carefully preprocessed.
2. **Model Complexity and Overfitting:**
    - ANN and LSTM models showed **a drop in validation accuracy compared to training accuracy**, suggesting overfitting.
    - GNN models require careful **graph construction and feature selection** to avoid overfitting due to high model complexity.
3. **Scalability and Computational Cost:**
    - **LSTM and RNN require extensive training time** due to sequential data dependencies, making them computationally expensive.
    - **GNNs require graph construction** and feature extraction steps that can be computationally intensive, particularly for large datasets.
4. **Error in Q8 Predictions:**
    - The **LSTM model had higher MAE in Q8 prediction**, indicating difficulty in predicting 8-state structures compared to the simpler Q3 task.
    - The **higher complexity of Q8 labels** leads to more ambiguity and potential misclassification.
5. **Data Preprocessing Constraints:**
    - **Sequence filtering** (removing sequences longer than 128) may exclude important information.
    - **Padding strategies** in sequence-based models can influence performance, especially in RNN and LSTM.

The GNN model emerged as the most effective approach, achieving the highest accuracy 90% for Q3 prediction. RNN performed well with an accuracy of 83.77%, making it a suitable alternative when graph-based modeling is not feasible. LSTM models, though effective for Q8 classification, exhibited higher error rates, suggesting room for optimization. Future work should focus on improving encoding techniques, reducing overfitting, and optimizing GNN models for better computational efficiency.

Our method, faces some inherent challenges even as there have been great strides in protein structure prediction. In the first place, although more sophisticated machine learning architectures like Graph Neural Networks, attention mechanisms[16] and

ensemble learning have generally improved accuracy, such approaches highly rely on the accurate and available set of evolutionary and co-evolutionary data. However, if rarely observed structural motifs or minority protein classes are to be anticipated, then this reliance can become particularly problematic.

Second, hybrid physics-AI approaches like molecular dynamics relaxation and energy-based scoring functions adapt predicted conformations by correcting local errors and steric clashes. But these approaches introduce heavy calculations and may still not fully capture the intricate conformational dynamics observed in some proteins.

Moreover, while data augmentation and transfer learning strategies alleviate the problems of data scarcity, they add additional levels of complexity and, if not properly engineered, can propagate systematic biases [5]. Though appropriate in detecting regions of low confidence, uncertainty quantification methods do not entirely eliminate the differences but instead treat them as discrepancies that arise between model and computational observables and experimental observable in highly flexible or ambiguous protein regions.

Lastly, iterative active learning and benchmarking by community challenges, CASP[21], have identified recurrent lack of predictive capability for challenging interfaces and unambiguous structural details impacting systems like antibody–antigen interactions. Even minor improvements in accuracy are important, as they translate into significant cost savings and more efficient R&D in high-stakes applications.

Chapter 7

Future Work

Transformers and Graph Neural Networks (GNNs) have become potent tools in the field of protein structure prediction. The amalgamation of both models capitalizes on the advantages of both architectures: the capability of Transformers to capture long-range correlations through attention mechanisms, and the ability of GNNs to handle graph-structured input. Protein structure predictions might become more reliable and accurate with the use of this comprehensive technique. Since proteins are naturally represented as graphs, GNNs are especially well suited for protein structure prediction. Amino acids are represented as nodes in these graphs, and the spatial or sequential interactions between them are shown as edges. GNNs are particularly good at identifying local and global relationships in such graph-structured data. They use intricate relational structures and iterative data aggregation from neighboring nodes to provide detailed predictions of local structural properties such as secondary structures or contact maps. Regard-less of distance between inputs, transformers can easily find the dependencies involved in natural language processing. By making use of a self-attention process in which all pairs of symbols have equal weight, it thus represents any potential dependency for time-lagged signals. Here we find that transformers are useful in capturing long-range interactions so fundamental to the problem of protein structure prediction. Because of the self-attention process, a model can describe complex relationships that were missed by GNNs in the previous sentence. It allows for relative evaluation and importance compared to any one amino acid of each amino acid.

GNN models combined with large scale protein structure datasets are trained. The model learns by making the empirical structures, as well as the anticipated structure of these models, similar. It is possible to further improve the accuracy of the model by transfer learning or gently recalibrating specific protein families. When there is little high-quality training data, artificial mutations in combination with methods for generating various conformational states may be used to address this lack of high-quality training data. The self-attention mechanism computes the output representation for every element in the sequence by looking at all others in the sequence.

Chapter 8

Conclusion

In this paper, we adopt various deep learning architectures, including Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks and Graph Neural Networks (GNNs), to evaluate their effectiveness for predicting secondary structure The results show that RNNs perform better than ANNs because they can capture the relevance of amino acid chains RNNs achieved a Q3 accuracy of 83 77%, while LSTMs performed strongly in Q8 classification with an accuracy of 79 59%. The GNN-based approach proposed a new feature extraction method, transforming sequences into n-grams, which improved classification but still needs to be optimized further GNN achieved strong performance in Q3 classification, with an accuracy of 90%. Moreover, the stacked classifiers yielded low error rates, thereby confirming the robustness of these predictive models Our study underscored that deep-learning models make structural prediction feasible and less reliant on expensive experimental methods such as X-ray crystallography or NMR spectroscopy. While RNN and LSTM models proved highly effective, integrating graph-based approaches and hyperparameter tuning can further improve performance. Future research should focus on enhancing dataset diversity, optimizing feature extraction techniques, and exploring hybrid architectures to refine secondary structure prediction accuracy.

Bibliography

[1] T. B. Alakus and I. Turkoglu. "A novel protein mapping method for predicting the protein interactions in COVID-19 disease by deep learning". In: *Interdisciplinary Sciences: Computational Life Sciences* 13 (2021), pp. 44–60. URL: https://doi.org/10.1007/s12539-020-00405-4.

[2] T. B. Alakus and I. Turkoglu. "Prediction of Protein-Protein Interactions with LSTM Deep Learning Model". In: (2019). URL: https://ieeexplore.ieee.org/abstract/document/8932876.

[3] H. Alquran et al. "A comprehensive framework for advanced protein classification and function prediction using synergistic approaches: Integrating bispectral analysis, machine learning, and deep learning". In: *Plos One* 18.12 (2023), e0295805. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0295805.

[4] T. Bepler and B. Berger. "Learning the protein language: Evolution, structure, and function". In: *Cell Systems* 12.6 (2021), pp. 654–669. URL: https://www.cell.com/cell-systems/pdf/S2405-4712(21)00203-9.pdf.

[5] Xiao Cao et al. "PSSP-MVIRT: peptide secondary structure prediction based on a multi-view deep learning architecture". In: *Briefings in Bioinformatics* 22.6 (June 2021), bbab203. ISSN: 1477-4054. DOI: 10.1093/bib/bbab203. eprint: https://academic.oup.com/bib/article-pdf/22/6/bbab203/41089195/bbab203.pdf. URL: https://doi.org/10.1093/bib/bbab203.

[6] E. Cappetta, G. Andolfo, and A. et al. Guadagno. "Tomato genomic prediction for good performance under high-temperature and identification of loci involved in thermotolerance response". In: *Horticulture Research* 8 (2021). URL: https://academic.oup.com/hr/article/doi/10.1038/s41438-021-00647-3/6491082.

[7] T. R. Chen et al. "A Secondary Structure-Based Position-Specific Scoring Matrix Applied to the Improvement in Protein Secondary Structure Prediction". In: *PLoS ONE* 16.7 (2021), e0255076. DOI: 10.1371/journal.pone.0255076. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0255076.

[8] Y. Chen. "Prediction of Protein Secondary Structure using SVM-PSSM Classifier Combined by Sequence Features". PhD thesis. Your University Name, 2016. URL: https://ieeexplore.ieee.org/abstract/document/7867181.

[9] Jinyong Cheng, Ying Xu, and Yunxiang Zhao. "Prediction of protein secondary structure based on deep residual convolutional neural network". In: *Biotechnology & Biotechnological Equipment* 35.1 (2021), pp. 1881–1890.

[10] Hsin-Hung Chou. *Protein PDB Dataset*. 2021. DOI: 10.21227/b02c-1p09. URL: https://dx.doi.org/10.21227/b02c-1p09.

[11] Giovanni Corsetti et al. "Importance of Energy, Dietary Protein Sources, and Amino Acid Composition in the Regulation of Metabolism: An Indissoluble Dynamic Combination for Life". In: *Nutrients* 16.15 (2024). ISSN: 2072-6643. DOI: 10.3390/nu16152417. URL: https://www.mdpi.com/2072-6643/16/15/2417.

[12] V. Enireddy, C. Karthikeyan, and D. V. Babu. "One-Hot Encoding and LSTM-Based Deep Learning Models for Protein Secondary Structure Prediction". In: *Soft Computing* 26.8 (2022), pp. 3825–3836. DOI: 10.1007/s00500-022-06783-9. URL: https://link.springer.com/article/10.1007/s00500-022-06783-9.

[13] N. Giri, R. S. Roy, and J. Cheng. "Deep learning for reconstructing protein structures from cryo-EM density maps: Recent advances and future directions". In: *Current Opinion in Structural Biology* 79 (2023), p. 102536. URL: https://doi.org/10.1016/j.sbi.2023.102536.

[14] Y. Guo et al. "Bagging MSA Learning: Enhancing Low-Quality PSSM with Deep Learning for Accurate Protein Structure Property Prediction". In: *Research in Computational Molecular Biology: 24th Annual International Conference, RECOMB 2020*. Vol. 12074. Lecture Notes in Computer Science. Springer International Publishing, 2020, pp. 88–103. DOI: 10.1007/978-3-030-45257-5_6. URL: https://link.springer.com/chapter/10.1007/978-3-030-45257-5_6.

[15] Yanbu Guo et al. "DeepACLSTM: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction". In: *BMC bioinformatics* 20 (2019), pp. 1–12.

[16] Ayush Hemnani and Nagamma Patil. "Enhanced Protein Secondary Structure Prediction with Bidirectional LSTM and Attention Mechanism". In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2024, pp. 1–6. DOI: 10.1109/ICCCNT61001.2024.10724878.

[17] D. P. Ismi and R. Pulungan. "Deep Learning for Protein Secondary Structure Prediction: Pre- and Post-AlphaFold". In: *Computational and Structural Biotechnology Journal* 20 (2022), pp. 6271–6286. DOI: 10.1016/j.csbj.2022.11.011. URL: https://www.sciencedirect.com/science/article/pii/S2001037022005062.

[18] Dewi Pramudi Ismi, Reza Pulungan, et al. "Deep learning for protein secondary structure prediction: Pre and post-AlphaFold". In: *Computational and structural biotechnology journal* 20 (2022), pp. 6271–6286.

[19] J. et al. Jumper. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (2021), pp. 583–589. URL: https://doi.org/10.1038/s41586-021-03819-2.

[20] S. Khan et al. "PSSM-Sumo: Deep Learning-Based Intelligent Model for Prediction of Sumoylation Sites Using Discriminative Features". In: *BMC Bioinformatics* 25.1 (2024), p. 284. DOI: 10.1186/s12859-024-05917-0. URL: https://link.springer.com/article/10.1186/s12859-024-05917-0.

[21] Andriy Kryshtafovych et al. “Critical assessment of methods of protein structure prediction (CASP)—Round XV”. In: *Proteins: Structure, Function, and Bioinformatics* 91.12 (2023), pp. 1539–1549. DOI: https://doi.org/10.1002/prot.26617. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26617. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26617.

[22] V. Marx. “Method of the Year: protein structure prediction”. In: *Nature Methods* 19 (2022), pp. 5–10. URL: https://doi.org/10.1038/s41592-021-01359-1.

[23] B. Mewara and S. Lalwani. “Sequence-based prediction of protein–protein interaction using auto-feature engineering of RNN-based model”. In: *Research in Biomedical Engineering* 39 (2023), pp. 259–272. URL: https://doi.org/10.1007/s42600-023-00273-z.

[24] A. M. Sequeira, D. Lousa, and M. Rocha. “ProPythia: a Python package for protein classification based on machine and deep learning”. In: *Neurocomputing* 484 (2022), pp. 172–182. URL: https://doi.org/10.1016/j.neucom.2021.07.102.

[25] Jaspreet Singh et al. “SPOT-1D-Single: improving the single-sequence-based prediction of protein secondary structure, backbone angles, solvent accessibility and half-sphere exposures using a large training set and ensembled deep learning”. In: *Bioinformatics* 37.20 (2021), pp. 3464–3472.

[26] K. Stahl, A. Graziadei, and T. et al. Dau. “Protein structure prediction with in-cell photo-crosslinking mass spectrometry and deep learning”. In: *Nature Biotechnology* (2023). URL: https://doi.org/10.1038/s41587-023-01704-z.

[27] M. Torrisi. “Predicting Protein Structural Annotations by Deep and Shallow Learning”. In: (2020). URL: https://researchrepository.ucd.ie/handle/10197/12811.

[28] S. Wang, J. Peng, and J. Xu. “Protein Secondary Structure Prediction Using Deep Learning Model”. In: *BMC Bioinformatics* 20 (2019). DOI: 10.1186/s12859-019-3119-7. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-7.

[29] Sheng Wang et al. “DeepCNF-D: predicting protein order/disorder regions by weighted deep convolutional neural fields”. In: *International journal of molecular sciences* 16.8 (2015), pp. 17315–17330.

[30] L. Yuan et al. “DLBLS_SS: Protein Secondary Structure Prediction Using Deep Learning and Broad Learning System”. In: *RSC Advances* 12.52 (2022), pp. 33479–33487. DOI: 10.1039/D2RA06433B. URL: https://pubs.rsc.org/en/content/articlehtml/2022/ra/d2ra06433b.

[31] Buzhong Zhang, Jinyan Li, and Qiang Lü. “Prediction of 8-state protein secondary structures by a novel deep learning architecture”. In: *BMC bioinformatics* 19 (2018), pp. 1–13.

[32] M. Zubair et al. “A Deep Learning Approach for Prediction of Protein Secondary Structure”. In: *Computers, Materials & Continua* 72.2 (2022), pp. 3705–3718. DOI: 10.32604/cmc.2022.025251. URL: https://www.techscience.com/cmc/v72n2/47234.