A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Supervisor:

Nazia Farah Signature

Lecturer

Department of Computer Science and Engineering

North Western University

Khulna, Bangladesh

**Acknowledgement**

All the praise to the almighty God, whose blessing and mercy succeeded me to complete this thesis work fairly. After that, we humbly acknowledge the valuable suggestions, advice, guidance and sincere co-operation of Nazia Farah , Lecturer of Computer Science and Engineering, North Western University,under whose supervision this work was carried out. Her intellectual advice, encouragement and guidance make us feel confident and scientific research needs much effort in learning and applying and need to have a broad view at problems from different perspective. We would like to convey our heartiest gratitude to all the faculty members, official and staffs of the Department of Computer Science and Engineering as they have always extended their cooperation to complete this work. Last but not least, we wish to thank our friends and family members for their constant supports.

**Abstract:**

Protein secondary structure predictions is an emerging topic nowdays. Despite of few rare exceptions, all proteins made of with the the same 20 amino acids. Protein's differ from each other in their shape. As protein secondary structure provides valuable insight into the three-dimensional formation of a protein by which we can also know the functionality of a protein. This prediction also tell us about the interaction, specificity, stability of protein with other molecules. Here in our study, we introduced an Ensemble model to predict protein secondary structure. We combined different machine learning models such as graph neural networks (GNNs), bidirectional LSTMs (Bi-LSTMs), and simple LSTMs (LSTMs) to build an ensemble model that improved the accuracy of protein secondary structure prediction.After preprocessing we also applied Random Forest(RF),Decision Tree(DT), Gradient Boost(GB) and k-Nearest-Neighbour(KNN) classifier to make prediction.Then we also applied ensemble model with the Random Forest(RF), Decision Tree(DT),k-Nearest-Neighbour(KNN) classifier. In our study we applied Voting Ensemble two times that follows the Majority Rule.We applied orthogonal one-hot encoding to make idiosyncratic binary vectors which corresponds to the each of the twenty amino acids.As amino acids in the primary structure are connected to two neighbor amino acids,depends on the interconnected sequence we made a graph for our GNN model.In GNN model the feature vector of each node is the one-hot-encoding binary vector.Then we iterate the entire graph to aggregate the information of the neighbouring nodes.To predict and using Bi-LSTM and LSTM we used numeric encoding of the corresponding amino acids.Then we applied our proposed bi-lstm and lstm model for prediction. After all we applied Voting Ensemble model on the GNN,Bi-LSTM and LSTM model to get better accuracy.We also compared those models performance on RSCB PDB dataset. Our Voting Ensemble model outperformed all the existing models with accuracy 95.12%.

The prediction of protein three-dimensional structure from amino acid sequence has been a grand challenge problem in computational biophysics for decades, owing to its intrinsic scientific interest and also to the many potential applications for robust protein structure prediction algorithms, from genome interpretation to protein function prediction. More recently, the inverse problem — designing an amino acid sequence that will fold into a specified three-dimensional structure — has attracted growing attention as a potential route to the rational engineering of proteins with functions useful in biotechnology and medicine. Methods for the prediction and design of protein structures have advanced dramatically in the past decade. Increases in computing power and the rapid growth in protein sequence and structure databases have fueled the development of new data-intensive and computationally demanding approaches for structure prediction. New algorithms for designing protein folds and protein–protein interfaces have been used to engineer novel high-order assemblies and to design from scratch fluorescent proteins with novel or enhanced properties, as well as signaling proteins with therapeutic potential. In this Review, we describe current approaches for protein structure prediction and design and highlight a selection of the successful applications they have enabled.

**Contents**

Acknowledgement ……………………………………………………

Abstract ……………………………………………………………

List of Figures ……………………………………………………….

List of Tables ……………………………………………………….

List of Abbreviations ………………………………………………..

**TABLE OF CONTENTS**

**Topic; Page:**

**LIST OF TABLES ……………………………………………………………………..**

**LIST OF FIGURES ……………..……………………………………………………..**

**CHAPTER ONE: INTRODUCTION ………………………………………………..**

**CHAPTER TWO: PROTEIN ……….………………………………………………..**

**CHAPTER THREE: PROTEIN STRUCTURE ...…………………………………..**

**3.1 The 4 Levels of Protein Structure …………………………………………**

**3.2 Protein Primary Structure …………………………………………………**

**3.3 Protein Secondary Structure ……………………………………………….**

**3.4 Protein Tertiary Structure …………………………….……………………**

**3.4.1 The 4 Types of Protein Tertiary Structure ……………………………..**

**Ionic bonds**

**Disulfide bridges**

**Hydrophobic forces**

**Hydrogen bonds**

**3.5 Protein Quatemary Structure ……………………………………………**

**3.6 Protein structure prediction used for- ……………………………………**

**CHAPTER FOUR: PROTEIN-FOLDING FORCES ………………………………**

**CHAPTER FIVE: TEMPLATE-BASED MODELLING……………….…………..**

**5.1 Template-Free Modelling ...…………………………………………………**

**LIST OF FIGURES**

**Figure No. Description Page**

1 The 4 Levels of Protein Structure **……………….…………………**

**2** Protein Primary Structure …………………………………………

**3**  Protein Secondary Structure ………………………………………

**4**  Protein Tertiary Structure …………………………………………

**5**  Protein Quaternary Structure ……………………………………...

**6** The 4 Types of Protein Tertiary Structure ……..…………………..

**7** Protein-Folding Forces …………………………..………………...

**8** Template-Free Modelling ………………………..…………………

**List of Abbreviations**

GNN Graph Neural Network

SVM Support Vector Machine

PSSP Protein Secondary Structure Prediction

DNN Deep Neural Network

DSSP Dictionary of Protein Secondary Structure

PDB Protein Data Bank

PSS Protein Secondary Structure

Bi-LSTM Bidirectional Long Short-Term Memory

LSTM Long Short-Term Memory

RF Random Forest

DT Decision Tree

KNN K-Nearest-Neighbour

GB Gradient Boost

**CHAPTER ONE: INTRODUCTION**

Proteins are essential biomolecules that play critical roles in various cellular processes. The structural arrangement of proteins, known as protein secondary structure, refers to the local spatial arrangement of amino acid residues within a protein chain, including alpha helices, beta sheets, and coil regions. Accurate prediction of protein secondary structure is vital for understanding protein function, dynamics, and interactions, and it has significant implications in drug discovery, protein engineering, and bioinformatics. In recent years, deep neural networks (DNNs) have emerged as powerful tools for protein secondary structure prediction. DNNs, a class of machine learning models, can automatically learn complex patterns and representations from large datasets, making them highly capable of capturing the intricate relationships between amino acid sequences and their corresponding secondary structures. Various types of DNNs, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their variants, have been employed to predict protein secondary structure with remarkable accuracy. However, despite their individual successes, single DNN models may have limitations, such as overfitting, lack of generalization, and sensitivity to hyper parameters. To address these challenges, researchers have turned to combining multiple DNN models to improve the accuracy and robustness of protein secondary structure prediction. The combination of DNNs can leverage the strengths of different architectures, exploit diverse features, and mitigate the weaknesses of individual models. There are several approaches for combining DNNs for protein secondary structure prediction, including ensemble methods, hybrid models, and deep stacking. Ensemble methods involve training multiple DNNs independently and combining their predictions to obtain a consensus prediction. Hybrid models integrate multiple types of DNNs or other machine learning algorithms into a single framework to exploit their complementary strengths. Deep stacking is a multi-layered approach that combines predictions from multiple DNN models to generate an ensemble prediction. Combining DNNs for protein secondary structure prediction has shown promising results, outperforming individual models and achieving state-of-the-art accuracy. However, it also poses challenges in terms of model selection, training data representation, and interpretability. Additionally, the interpretation of combined models and the understanding of their underlying mechanisms are still active areas of research. In conclusion, the combination of DNNs for protein secondary structure prediction is a promising approach to improve the accuracy and robustness of predictions. It has the potential to enhance our understanding of protein structure and function, and it holds great promise for advancing fields such as drug discovery, protein engineering, and bioinformatics.

The stunning diversity of molecular functions performed by naturally evolved proteins is made possible by their finely tuned three-dimensional structures, which are in turn determined by their genetically encoded amino acid sequences. A predictive understanding of the relationship between amino acid sequence and protein structure would therefore open up new avenues, both for the prediction of function from genome sequence data and also for the rational engineering of novel protein functions through the design of amino acid sequences with specific structures. The past decade has seen dramatic improvements in our ability to predict and design the three-dimensional structures of proteins, with potentially far-reaching implications for medicine and our understanding of biology. New machine-learning algorithms have been developed that analyse the patterns of correlated mutations in protein families, to predict structurally interacting residues from sequence information alone1,2. Improved protein energy functions3,4 have for the first time made it possible to start with an approximate structure prediction model and move it closer to the experimentally determined structure by an energy-guided refinement process5,6. Advances in protein conformational sampling and sequence optimization have permitted the design of novel protein structures and complexes7,8, some of which show promise as therapeutics9.These advances in protein structure prediction and design have been fueled by technological breakthroughs as well as by a rapid growth in biological databases. Protein-modelling algorithms (Box 1) are computationally demanding both to develop and to apply. The rapid increase in computing power available to researchers (both CPU-based and, increasingly, GPU-based computing power) facilitates rapid benchmarking of new algorithms and enables their application to larger molecules and molecular assemblies. At the same time, next-generation sequencing has fueled a dramatic increase in protein sequence databases as genomic and metagenomic sequencing efforts have expanded10. Advances in software and automation have increased the pace of experimental structure determination, speeding the growth of the database of experimentally determined protein structures (the Protein Data Bank (PDB))11, which now contains close to 150,000 macromolecular structures. Deep-learning algorithms12 that have revolutionized image processing and speech recognition are now being adopted by protein modelers seeking to take advantage of these expanded sequence and structural databases.

**Objectives**

In our study, we applied our proposed GNN, Bi-LSTM and LSTM models on the dataset respectively. In GNN model creation steps, initially we built the graph according to the input sequences. Then we applied neural network on the created graph. Here, for each node we sum up that node’s neighbor nodes information to that node. For the Bi-LSTM and LSTM approach, firstly we passed the padded the input data to the embedding layer, then bi-lstm/lstm layers, dense layer respectively. Then we applied softmax function in the output layer. For compilation we used "rmsprop" optimizer, and "categorical\_crossentropy" loss function. Lastly we applied Voting Ensemble model on the GNN, Bi-LSTM and LSTM results to avoid biasness of a model. Our Voting Ensemble model outperformed all the existing models with accuracy 95.12%. We also followed after a few epochs of running, the training loss decreased and the accuracy enhanced . This phenomenon indicated that our model have been trained perfectly on the dataset without over-fitting and under-fitting. During GNN implementation, a direct or indirect graph is used as input, which encodes data and then generates a target value based on graph node and edge information. Here used static undirected homogeneous graphs were applied using a supervised setup method among the various types of graphs.

**Motivation**

Proteins are enormous, complex molecules that play a variety of important tasks in the human body. Protein secondary structure is intertwined with protein tertiary structure, which dictates the proteins’ unique behavior. Many studies and predictions of protein structure have been conducted over the years. Although there are over a million known protein sequences, the total number of experimentally determined structures is less than 20,000 (Protein Data Bank). As a result, it’s becoming increasingly important to predict protein structure from its amino acid sequence, based on existing structures.

However, by utilizing computational power, we may generate a large number of secondary structures from a given primary structure in a shorter amount of time and effort. As technology advances, the accuracy of the prediction improves at a rapid rate, thanks to new technologies and varied methods.

Ensemble is a new combined approach based on different models. The application and exploration of Ensemble model in bioinformatics are minimal as a new model. To that end, we are inspired to investigate this cutting-edge theory in our thesis and apply it to benefit mankind and the globe.

**Contributions**

The main aim of our study is to develop an Ensemble Model to predict protein secondary structure. We proposed three models and these are Bi-Lstm, Lstm and GNN models. Then we applied Voting Ensemble model to get better accuracy. We also applied different classifier models for performance evaluation of the dataset.

The main contributions of the thesis are as follows:

* Apply our proposed GNN model and got the accuracy for that GNN model.
* Apply our proposed Bi-Lstm and Lstm model and got accuracy for both of the models.
* Apply Voting Ensemble model .
* Apply different types of classifier such as DT, KNN, AdaBoost, GardientBoost, RF and then apply ensemble model on these classifier.

**Methodology**

Firstly, we preprocessed our dataset based on the predefined maximum length and depending on the existence of non-standard amino acids.

At the first stage, on the preprocessed dataset we applied two types of encoding and these are: 1. One-hot encoding and 2. Numeric encoding

Here, we used one-hot encoding for GNN model and Numeric encoding for Bi-Lstm and Lstm models.

After that, we applied Voting Ensemble on the three models. Then we applied different types of classifier such as DT, RF, KNN, AdaBoost and GradientBoost.

Lastly, we applied Voting Ensemble model on the classifier models and compared those results.

**CHAPTER TWO: PROTEIN**

Protein is a large biological molecule that is essential to the structure, function, and regulation of cells, tissues, and organs within living organisms. It is composed of chains of amino acids that fold into specific 3D structures, determined by the sequence of amino acids, which is encoded by the genetic information in DNA. Proteins perform a wide variety of functions in the body, such as catalyzing biochemical reactions, serving as structural components of cells and tissues, transporting molecules across cell membranes, regulating gene expression, and acting as signaling molecules. Protein is an important macronutrient that is required for growth, maintenance, and repair of tissues in the body, and can be obtained through the diet by consuming protein-rich foods such as meat, eggs, dairy products, beans, and nuts.

**CHAPTER THREE: PROTEIN STRUCTURE**

Protein structure prediction is the inference of the three-dimensional structure of a protein from its amino acid sequence—that is, the prediction of its secondary and tertiary structure from primary structure. Structure prediction is different from the inverse problem of protein design. Structure prediction is the prediction of the three-dimensional structure of materials, such as crystals, proteins or small molecules. There are two general approaches to predicting the structure of a protein of interest (the ‘target’): template-based modelling, in which the previously determined structure of a related protein is used to model the unknown structure of the target; and template-free modelling, which does not rely on global similarity to a structure in the PDB and hence can be applied to proteins with novel folds. Historically, the methods applied in these two approaches have been quite distinct, with template-based modelling focusing on the detection of, and alignment to, a related protein of known structure, and template-free modelling relying on large-scale conformational sampling and the application of physics-based energy functions. Recently, however, the line between these approaches has begun to blur, as template-based methods have incorporated energy-guided model refinement, and template-free methods have employed machine learning and fragment-based sampling approaches to exploit the information in the structural database (although template-based methods still retain an increased accuracy for targets with detectable sequence similarity to the entries in the PDB). Here we provide a brief introduction to template-based modelling methods, and then turn to template-free modelling and describe recent developments in that areas.

3.1 The 4 Levels of Protein Structure:

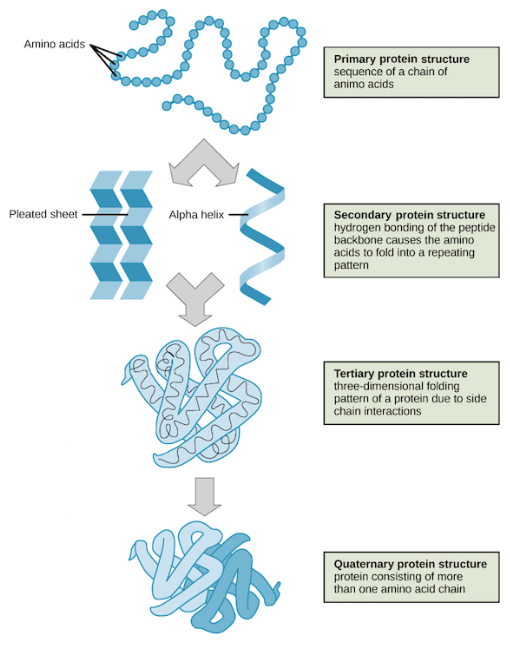
To understand how a protein gets its final shape or conformation, we need to understand the four levels of protein structure. The complete structure of a protein can be described at four different levels of complexity:

> Primary Structure

> Secondary Structure

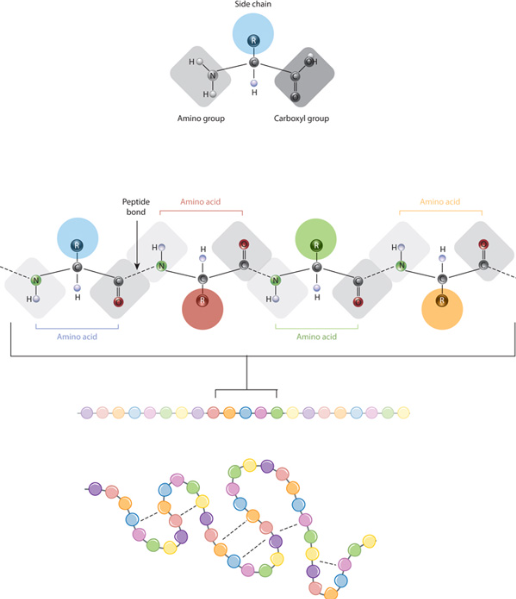
> Tertiary Structure

> Quaternary Structure



**3.2** **Protein Primary Structure**

To reiterate, the primary structure of a protein is defined as the sequence of amino acids linked together to form a polypeptide chain. Each amino acid is linked to the next amino acid through peptide bonds created during the protein biosynthesis process. Proteins are built from a set of only twenty amino acids, each of which has a unique side chain. The side chains of amino acids have different chemistries. By convention, the primary structure of a protein is reported starting from the amino-terminal (N) end to the carboxyl-terminal (C) end. Protein biosynthesis is most commonly performed by ribosomes in cells. The specific sequence is very important, since even a small change (called a mutation) could cause a disorder.



Primary structure, the most basic level of protein sequence, is just the arrangement of amino acids in a polypeptide chain. A protein can have up to 20 different amino acids. As an illustration in the table 1.

Table 1:Symbol And Name Of Amino Acids

Index Symbol Name

1 A Alanine

2 R Arginine

3 N Asparagine

4 D Aspartate

5 C Cysteine

6 Q Glutamine

7 E Glutamate

8 G Glycine

9 H Histidine

10 I Isoleucine

11 L Leucine

12 K Lysine

13 M Methionine

14 F Phenylalanine

15 P Proline

16 S Serine

17 T Threonine

18 W Tryptophan

19 Y Tyrosine

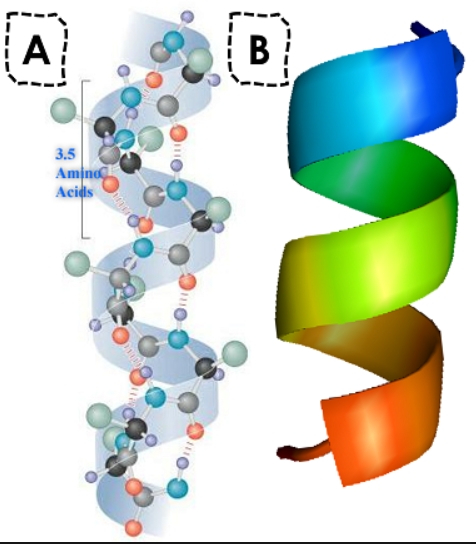
20 V Valine

The properties of the 20 amino acids can be used to classify them. Charge, hydrophilicity or hydrophobicity, size, and functional groups are all important considerations. Protein structure and protein-protein interactions are influenced by these features. The linear combination of amino acids that will be used as input to the proposed model is known as protein primary

structures.

**3.3** **Protein Secondary Structure**

The secondary structure arises from the hydrogen bonds formed between atoms of the polypeptide backbone. The hydrogen bonds form between the partially negative oxygen atom and the partially positive nitrogen atom. Localized structures that form based on interactions within the protein backbone. The most common types of secondary structures are the α helix and the β pleated sheet. Both structures are held in shape by hydrogen bonds, which form between the carbonyl and amino group of the peptide bond the secondary structure of proteins is marked by specific regions of the polypeptide form set very distinct structures. The two most common shapes that appear at this stage are the alpha helix (α-helix) and the beta-pleated sheet (β-pleated sheet). A secondary structure of a protein pertains to the folding of a polypeptide chain, resulting in an alpha helix, beta sheet or a random coil structure. Another example of a secondary structure is that of a nucleic acid such as the clover leaf structure of tRNA.



During the secondary and territory structure generation, protein structures can always define 3 structures. Since each amino acid molecule can be in one of three states: alpha helices, beta sheets, and turns, these are referred to as 3-state amino acids. With all these kinds of singleletter symbols, the Dictionary of Protein Secondary Structure (DSSP) codes are commonly employed to illustrate and depict the protein secondary structure[11].

Furthermore, these three states extend into eight states, as seen in the table 2.

Table 2: Secondary Structure Description

Sr. No. Code Description

1 H alpha helix

2 G 310 helix

3 I pi helix

4 B beta bridge

5 E extended beta sheet

6 C coil

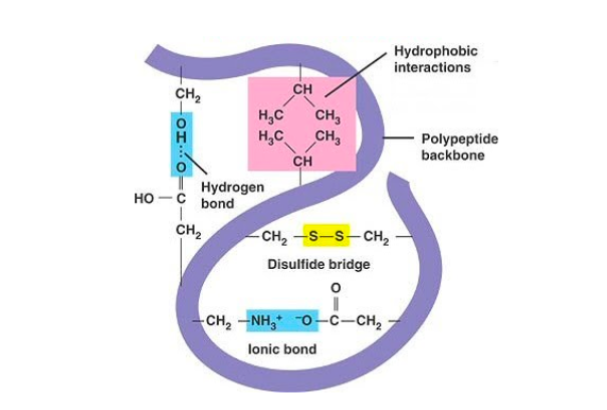
7 S bend

8 T hydrogen bonded turn

**3.4 Protein Tertiary Structure**

The tertiary structure of a protein refers to the overall three-dimensional arrangement of its polypeptide chain in space. It is generally stabilized by outside polar hydrophilic hydrogen and ionic bond interactions, and internal hydrophobic interactions between nonpolar amino acid side chains. The polypeptide chain may experience additional folding to produce the tertiary structure. To some extent, the tertiary structure is determined by the amino acid sequence of the primary structure. For example, in globular proteins, polypeptide chains are held together in a certain way to form a compact structure. Tertiary Structure Deals with the Three-Dimensional Arrangement of All of the Amino acids. The tertiary structure of proteins deals with how the regional structures are put together in space. For example, the α-helices may be oriented parallel to each other or at right angles. A protein’s tertiary structure is frequently split into domains, which are separate compact folding units that typically contain 100 to 200 residues. Small proteins may only have one domain, however complex proteins frequently have several. A fold is a typical feature of many proteins that refers to the spatial assembling of secondary structural elements into a domain-like structure. Folds are frequently, but not always, associated with a particular function.

A structural motif is comparable to a fold, although it is usually smaller and serves as the foundation for folds. Several structural motifs can be found in a wide range of irrelevant proteins with various changes, whereas others, particularly those associated with a specific biochemical function, are structurally similar and commonly found in protein domains with comparable characteristics.



**3.4.1 The 4 Types of Protein Tertiary Structure**

The tertiary structure involves four types of bonds:

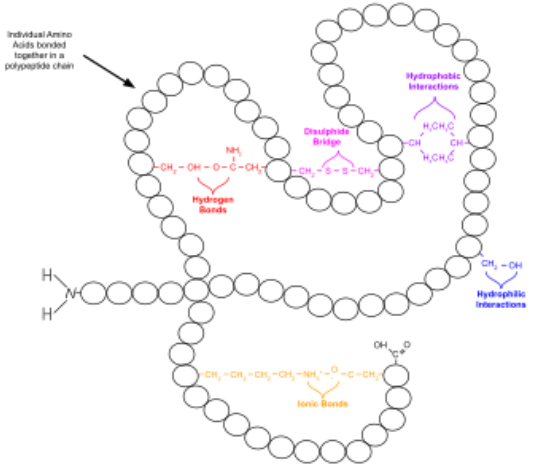
**> Ionic bonds**: Positively charged R groups bond together.

**> Disulfide bridges:** The amino acid cysteine forms a bond with another cysteine through its R group.

**> Hydrophobic forces:** These amino acids orient themselves towards the center of the polypeptide to avoid the water.

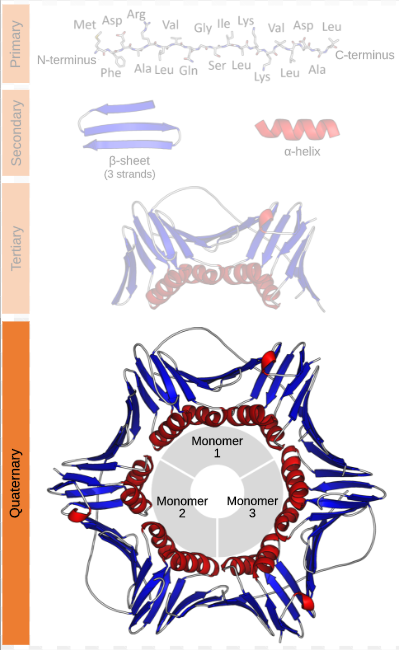
**> Hydrogen bonds:** Polar R groups on the amino acids form bonds with other Polar R groups.

The bonds in the tertiary structure of a protein involve disulfide bonds, hydrogen bonds, ionic bonds, and hydrophobic interactions. These bonds create the three-dimensional shape of a protein that gives it the tertiary.



**3.5 Protein Quatemary Structure**

The quaternary structure of a protein is the association of several protein chains or subunits into a closely packed arrangement. Each of the subunits has its own primary, secondary, and tertiary structure. Protein quaternary structure refers to the structure of proteins which are themselves composed of two or more smaller protein chains (also referred to as subunits). Protein quaternary structure describes the number and arrangement of multiple folded protein subunits in a multi-subunit complex. The subunits are held together by hydrogen bonds and van der Waals forces between nonpolar side chains. The quaternary structure refers to the number and arrangement of the protein subunits with respect to one another. Examples of proteins with quaternary structure include hemoglobin, DNA polymerase, ribosomes, antibodies, and ion channels. Quaternary structure in proteins is the most intricate degree of organization still considered a single molecule. To be considered to have quaternary structure, a protein must have two or more peptide chains forming subunits. The subunits can be different or identical, and in most cases they are arranged symmetrically. The configuration of a protein substance created by collisions between several polypeptide chains is known as a quaternary structure. An aliquot is a polypeptide chain that is made up of several polypeptide chains. Multiple subunits of the same specific protein can be found in quaternary structure proteins. They could also be made up of various subunits. Hemoglobin is an illustration of a quaternary structure protein. Hemoglobin is an iron-containing protein found in the blood that attaches oxygen molecules. It is made up of four components: two alpha and two beta subunits.



**3.6 Protein structure prediction used for-**

Having a protein structure provides a greater level of understanding of how a protein works, which can allow us to create hypotheses about how to affect it, control it, or modify it. For example, knowing a protein's structure could allow you to design site-directed mutations with the intent of changing function.

**Graph**

**A graph is a non-linear data structure made up of nodes (also known as vertices) and edges. A graph is a visual depiction of a collection of things in which some objects are linked together by edges. The points that connect the interconnected items are called vertices, and the ties that interconnect the vertices are designated edges.**

**A graph is formally defined as a pair of sets (V, E), where V denotes the set of vertices and E is the set of edges that connect the pairs of vertices. Take a look at the graph in the figure 3.3.**

**where,**

**nodes or vertices,**

**V = A,B,C,D,E and Edge, E = AC, BC, DC, CE**

**An array of nodes and a two-dimensional array of edges can be used to represent a graph. Graphs have been used to address real-world problems in which the problem area is represented as a network. Telephone networks, circuit networks, social networks, and protein**

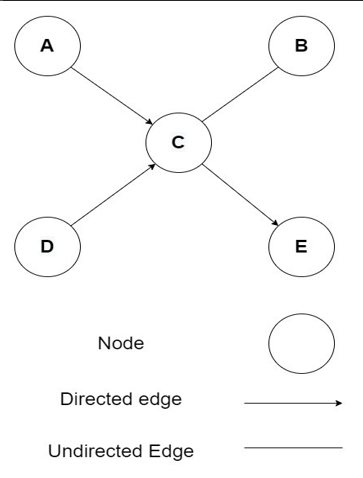
****

Figure: A Visual Representation of Graph.

**Vertex**

A vertex represents each node in the graph. The labeled circle in the figure 3.3 denotes vertices. As a result, A to E are vertices. An array can be used to represent them.

Edge

A path or a line connecting two vertices is represented by an edge. The lines from A to C, B to C, and so on indicate edges in the diagram 3.3. A two-dimensional array can be used to represent an edge.

Adjacency or Neighbour node

If two nodes or vertices are joined by an edge, they are considered a neighboring node or adjacent node. B is neighbor to C in the figure 3.3, while C is neighbor to {A, B, D, E}, and so on.

The most common graph structure scenarios are as follows:

• Structural Scenarios: It’s referred to as structural scenarios when graphs are explicit.

• Non-structural Scenarios: Non-structural circumstances occur where graphs are implicit and a graph is required to be built from a task.

Graphs come in a variety of shapes and sizes, and they may be viewed from a variety of perspectives. The graph can be customized based on weight.

• Weighted Graph - Graphs with values on their edges or routes. Weights refer to all of the values connected with the edges. The value of the edges can reflect weight, cost, or length.

• Unweighted Graph - When the edge has no value or weight attached to it. Unless there is a value connected with the graph, it is unweighted by default.

The following types of graphs can be classified based on their direction.

• Directed Graph - A digraph is a network of objects (node, edge) in which all of the vertices are routed from one vertex to the next.

• Undirected Graph - When a group of objects is linked together and all of their edges are bidirectional. In a connected two-node system, you can go in any direction from one to the other. Every edge in undirected graphs is considered as bi-directional edges.

Graphs can be categorized into the following categories based on their nodes and edges

similarity.

• Homogeneous Graph: Homogeneous graph’s edges and nodes are similar types. For example, a social network is a homogeneous graph made up of people and their connections that all reflect the same entity type.

• Heterogeneous Graph: Heterogeneous graph’s edges and nodes are not similar types. For example, buyer, product, and seller nodes are connected via wants-to-buy, iscustomer-of, has-bought, and is-selling edges in the graph encoding a marketplace.

Based on the runtime, graphs can be classified into the following categories.

• Static Graph: Static graph is effective for offline optimization/scheduling of graphs.

• Dynamic Graph: Dynamic graph’s input features vary with time which is more flexible.

To design the loss function, graph learning tasks should be realized. There are three categories of graph learning tasks.

• Node-level Tasks: Problems are solved in a semi-supervised way in node level tasks together with node classification (classify nodes into several classes), node regression (predicts a continuous value for every node), node clustering (divides the nodes into certain disjoint groups based on nodes’ types), etc.

• Edge-level Tasks: Link prediction and edge classification are considered edge-level tasks which need to predict existing edges between two given nodes or categorize edge types.

• Graph-level Tasks: All types of Graph-level tasks such as graph regression, graph classification, and graph matching require detection of graph representations.

In protein secondary structure prediction, the prediction has to make out of the primary sequence of a protein. After making the graph from the primary sequence(amino acid), the node level task is used to classify every node into any one of the 8 states of secondary structure.

From the viewpoint of supervision, three types of graph learning tasks based on training settings are given below.

• Supervised Setting: It supplies labeled data during training.

• Semi-supervised Setting: For training, it provides both unlabeled and labeled data where the ratio of unlabeled data is larger than labeled data. At the testing phase, the transductive situation supplies the prediction of labels of unlabeled vertices but the inductive situation produces new unlabeled vertices from a similar distribution. Lately, mixed setting such as mixed transductive-inductive scheme [15, 16] are implemented.

• Unsupervised Setting: To predict patterns of the model, it only provides unlabeled data.

LSTM

LSTM stands for Long Short-Term Memory, which is a type of recurrent neural network (RNN) architecture designed to overcome the vanishing gradient problem in traditional RNNs.

The vanishing gradient problem occurs when the gradients (i.e., the values used to update the model's parameters during training) become very small as they are propagated back through the layers of the network, which makes it difficult for the model to learn long-term dependencies.

LSTM addresses this problem by introducing a set of specialized memory cells that can selectively remember or forget information based on the input and the current state of the model. The memory cells allow LSTM to preserve information over long periods of time, making it well-suited for tasks that require processing sequences of inputs with long-term dependencies, such as speech recognition, language translation, and music generation.

LSTM is composed of several key components, including input gates, forget gates, output gates, and memory cells. These components work together to control the flow of information through the network and to update the state of the memory cells based on the input and the current state of the model.

Overall, LSTM is a powerful and versatile neural network architecture that has been widely used in a variety of applications, particularly in natural language processing and speech recognition.

BiLSTM Model

A Bidirectional Long Short-Term Memory (BiLSTM) model is a type of recurrent neural network (RNN) that has the ability to learn from sequences of data such as time series or natural language text.

In a standard LSTM, the input sequence is processed in a single direction, from the beginning to the end of the sequence. In contrast, a BiLSTM processes the input sequence in both forward and backward directions simultaneously, using two separate LSTM layers. This allows the model to capture information from both past and future context of each element in the input sequence.

The outputs from the forward and backward LSTMs are concatenated at each time step, producing a combined representation that is then used to make a prediction or generate the next step in the sequence. This combined representation enables the BiLSTM to capture complex relationships between the elements in the sequence, making it particularly useful for tasks such as sentiment analysis, speech recognition, and named entity recognition.

Overall, the BiLSTM model is a powerful tool for processing sequential data, thanks to its ability to capture both past and future context information.

GRU

GRU stands for "Gated Recurrent Unit," and it is a type of recurrent neural network architecture used in deep learning.

Recurrent neural networks (RNNs) are designed to handle sequential data, where the output of one step is used as input for the next step. However, traditional RNNs have a problem with vanishing gradients, which can make it difficult for the network to learn from long-term dependencies in the data.

GRUs were introduced in 2014 by Cho et al. as a solution to this problem. They are similar to traditional RNNs but have gating mechanisms that help the network to selectively remember or forget information from previous time steps, improving the ability of the network to capture long-term dependencies in the data.

A GRU cell has two gates - an update gate and a reset gate - that control the flow of information. The update gate determines how much of the previous state to keep and how much of the new state to add, while the reset gate determines how much of the previous state to forget.

GRUs have been shown to be effective in various tasks, such as speech recognition, language modeling, and machine translation, among others.

BiGRU

BiGRU stands for Bidirectional Gated Recurrent Unit, which is a type of neural network architecture used for processing sequential data such as text or speech.

A Bidirectional GRU is a variant of the Gated Recurrent Unit (GRU) model, which is a type of Recurrent Neural Network (RNN). RNNs are designed to handle sequential data by maintaining an internal state that allows them to remember past inputs. The GRU model is a variant of the more popular Long Short-Term Memory (LSTM) model and has fewer parameters, making it faster to train.

In a Bidirectional GRU, the input sequence is processed both forwards and backwards by two separate GRU layers. The outputs from the two layers are then concatenated, which allows the model to consider both past and future inputs when making predictions. This makes the Bidirectional GRU more powerful than a regular GRU model, particularly when dealing with complex sequential data where context is important.

Overall, BiGRU is a powerful and flexible neural network architecture that has been used in a variety of natural language processing (NLP) tasks such as text classification, sentiment analysis, and language translation.

RNN

RNN stands for "Recurrent Neural Network," a type of neural network designed to process sequential data, such as time series data or natural language sentences. Unlike traditional feedforward neural networks, which process inputs in a strictly linear manner, RNNs can process inputs with loops that allow information to persist across time.

The key feature of RNNs is that they have a "hidden state" that can be updated with each new input. This hidden state serves as a memory of what the network has seen so far, and it can be used to make predictions or classifications based on the entire sequence of inputs.

There are different types of RNNs, including Simple RNNs, LSTM (Long Short-Term Memory) RNNs, and GRU (Gated Recurrent Unit) RNNs. Each type has its own strengths and weaknesses, depending on the specific task at hand.

RNNs have been successfully applied in a variety of fields, including natural language processing, speech recognition, image captioning, and stock prediction. However, they can be challenging to train and prone to the problem of vanishing gradients, which can make it difficult for the network to retain long-term dependencies.

Bi RNN

A Bi-RNN, short for Bidirectional Recurrent Neural Network, is a type of neural network architecture that is commonly used in natural language processing and sequence modeling tasks.

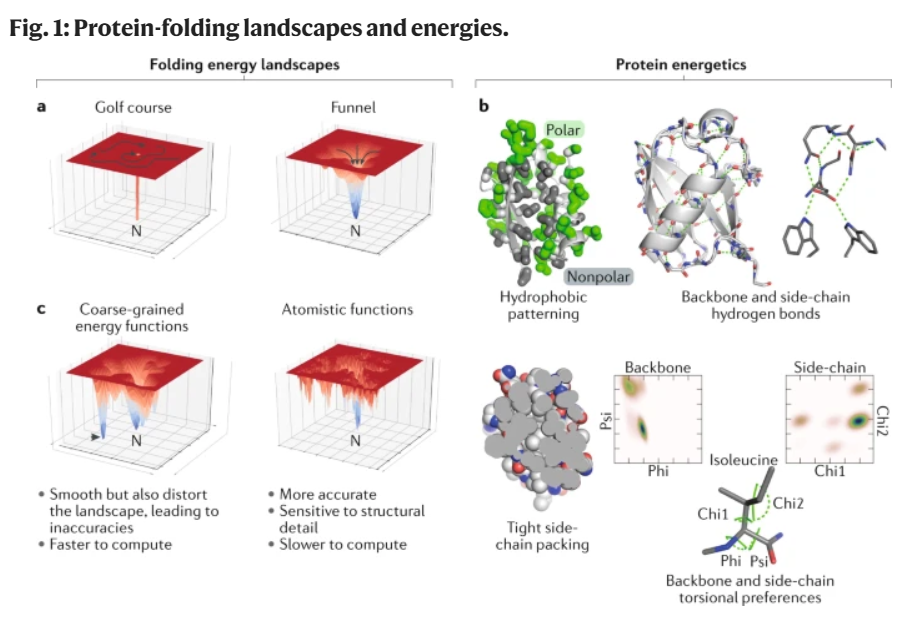
In a traditional recurrent neural network, information flows only in one direction, from the input sequence's beginning to its end. In contrast, a Bi-RNN consists of two recurrent neural networks that process the input sequence in opposite directions. The outputs of these two networks are then concatenated to create the final output.

The advantage of using a Bi-RNN is that it can capture information from both past and future states of the input sequence, which can be useful in tasks such as speech recognition or machine translation. By processing the input sequence in both directions, the Bi-RNN can learn to identify patterns and relationships that might not be apparent from processing the sequence in only one direction.

Overall, Bi-RNNs have become a popular choice for many sequence modeling tasks due to their ability to capture complex temporal dependencies in the data.

**CHAPTER FOUR: PROTEIN-FOLDING FORCES**

Proteins possess the remarkable ability to fold spontaneously into precisely determined three-dimensional structures. Refolding experiments have established that the information required to specify a protein’s folded conformation (its native state) is completely contained in its linear amino acid sequence13,14,15. According to Anfinsen’s thermodynamic hypothesis, this information is encoded in the shape of the energy landscape of the polypeptide: the native state is the one with the lowest free energy16,17. This hypothesis forms the basis for a general approach to protein structure prediction that combines sampling of alternative conformations with scoring to rank them by energy and identify the lowest energy state18,19,20,21. The chief obstacle to the success of this energy-guided approach, first identified by Cyrus Levinthal as a conceptual barrier to protein folding on biological timescales22, is the vast space of potential conformations: even supposing that each amino acid has only a limited, discrete set of possible backbone states, the total size of the conformational space that must be searched grows exponentially with chain length, and very quickly becomes astronomical. The solution to this dilemma lies in the recognition that it is not necessary to explore the entire conformational space in order to identify the native state: the energy landscape is not a flat ‘golf course’ with a single native ‘hole’; rather, directional cues impart an overall funnel shape to the landscape and guide sampling towards near-native conformations19,23 (Fig. 1a). These directional cues can arise from sequence-local residue interactions that bias short stretches of the chain towards forming specific secondary structures, or from favorable long-range, non-local packing interactions that can be formed even before the global native fold is reached.



**a |** Simplified, two-dimensional representations of ‘golf course’ and ‘funnel’-shaped energy landscapes. Identifying the native energy minimum (‘N’) in the landscape on the left requires exhaustive exploration, whereas a simple downhill search from most starting points will locate the native state in the landscape on the right.

**b |** Energetic features that distinguish the protein native state include: hydrophobic patterning (shown here in a cutaway view of the small protein ubiquitin), with burial of nonpolar side chains in the protein core; backbone and side-chain hydrogen bonding (hydrogen bonds are shown as dotted green lines); tight side-chain packing (visible in a slice through a protein core); and restricted backbone and side-chain torsion angle distributions (evident in the highly focused two-dimensional probability distributions of backbone — phi angle versus psi angle — and side-chain — chi1 angle versus chi2 angle — torsion angles for the amino acid isoleucine).

**c |** Computational models of protein energetics offer a trade-off between speed and accuracy. Coarse-grained models are computationally efficient and effectively smooth the energy landscape, permitting large-scale sampling; however, they also introduce inaccuracies such as false minima (for example, the blue basin to the left of the native minimum in this part, highlighted with an arrow). High-resolution, atomically detailed energy functions are more accurate, but also slower to evaluate and sensitive to structural detail, which introduces bumpiness (many local minima) into the landscape and makes them harder to navigate efficiently.

The driving force favoring the folding of water-soluble, globular proteins is thought to be the burial of hydrophobic amino acid side chains away from water24; folding is opposed by the loss of configurational entropy that accompanies the collapse of a flexible polypeptide chain into a defined 3D conformation. Tight packing of nonpolar side chains in the protein core enhances attractive van der Waals interactions and eliminates entropically unfavorable internal cavities (Fig. 1b). Moreover, this jigsaw puzzle-like packing is achieved while accommodating strong backbone and side-chain torsional preferences that restrict the observed torsion angle distributions (lower panels in Fig. 1b), effectively reducing side-chain flexibility to the neighborhood of a discrete set of rotamers at each position. Intra-protein hydrogen bonds and salt bridges largely compensate for the loss of interactions with water, as polar groups are buried during folding and hence these interactions contribute less to the stability of the native state than to its specificity (that is, they help discriminate the native state from other compact states). Whereas hydrophobic burial and backbone hydrogen bonding can be detected from low-resolution structural models, the tight core packing and absence of buried, unsatisfied polar groups that distinguish the native state require explicit modelling of the side-chain degrees of freedom. As a result, molecular modelling approaches for structure prediction and design often employ multiple levels of resolution: large-scale conformational sampling is performed with a computationally efficient coarse-grained energy function that captures hydrophobic burial, formation of secondary structure, and avoidance of atomic overlaps25,26,27; final protein model selection and refinement requires explicit modelling of the amino acid side chains using a more time-intensive, high-resolution atomistic energy function (Fig. 1c).

**Support Vector Machine**

SVM stands for support vector machine and is a classification approach that uses optimal margins. Object recognition,[17] speaker identification,[18] and text categorization[19] are just a few of the disciplines where SVM has been successfully utilized. SVM offers two primary advantages over newer algorithms: faster processing and better performance with fewer samples (in the thousands). When there are more dimensions than samples, SVM is effective. SVM’s core strategy entails transforming the samples into a high-dimensional Hilbert space and searching for a separating hyperplane inside it[20]. By adjusting the labeled training samples for each category, SVM aims to find an ideal detaching linear decision border or hyperplane with the biggest margin between the classes[21].

There are two types of SVM:

• Linear SVM: For data that is linearly separable (data that can be classified into two groups using only a single straight line), linear SVM is applied. An example can be used to explain how the Linear SVM algorithm performs. Let’s consider the following

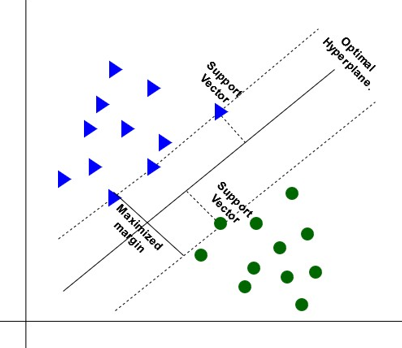


Figure : Visualization of Linear SVM.

figure , which contains a dataset with two tags (green and blue) and two features (x1 and x2). We’re seeking a classifier that can categorize the pair of coordinates (x1, x2) as green or blue. As this is a two-dimensional space, we can easily separate these two classes by simply drawing a straight line between them. However, numerous lines can be used to separate these classes. As a result, the SVM method aids in the discovery of the best line or decision boundary, which is referred to as a hyperplane. The nearest point (support vectors) of the lines from both classes is identified by using the SVM method. Margin is defined as the distance between the vectors and the hyperplane. The optimal hyperplane is the one with the largest margin.

• Non-linear SVM: For data that is non-linearly separable (data that cannot be classified into two groups using only a single straight line), Non-linear SVM is applied. Here,

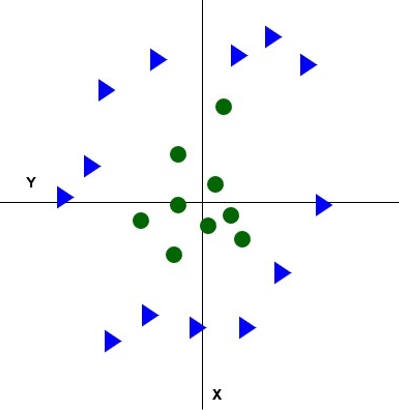
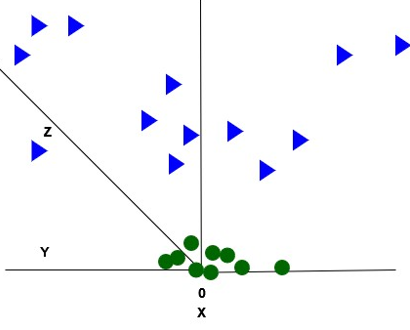


Figure : Visualization of Non-linear Dataset.

We are unable to draw even a single straight line (figure ). So we’ll need to add



**Figure :** Visualization of Non-linear Dataset in 3-D.

another dimension to separate these data points. We used two dimensions for linear data (x,y), therefore we’ll add a third dimension (z), for non-linear data (figure 2.4). It can be computed as follows:

*z* = *x*2 +*y*2

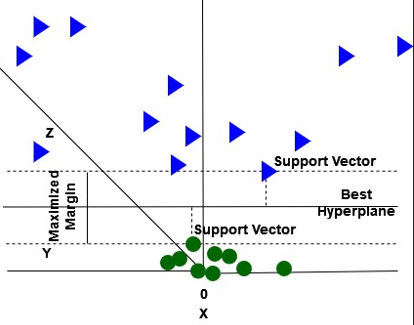


Figure : Visualization of Non-linear SVM In 3-D.

As seen in the following figure , SVM will now partition the datasets into classes. It appears to be a plane parallel to the x-axis as we’re in 3-d space (figure ). After

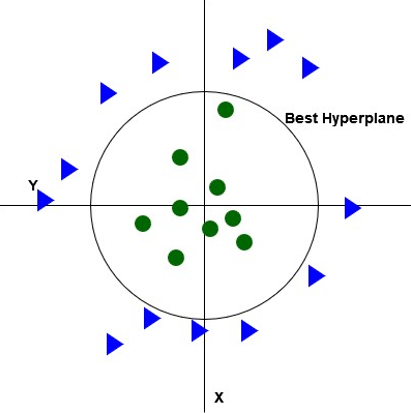


Figure : Visualization of Non-linear SVM.

converting it to 2d space with z=1, it looks like this (figure ), with radius=1.

SVM classifies a class by K(X, Xi) where K(X, Xi) is the subsequent kernel function:

SVM kernels include radial basis function (RBF), linear, sigmoid, and polynomial, among others [22].

• Radial Basis Function (RBF) Kernel: The radial basis function, which creates a "bump" around each data point, is one of the most widely used kernel functions.

K

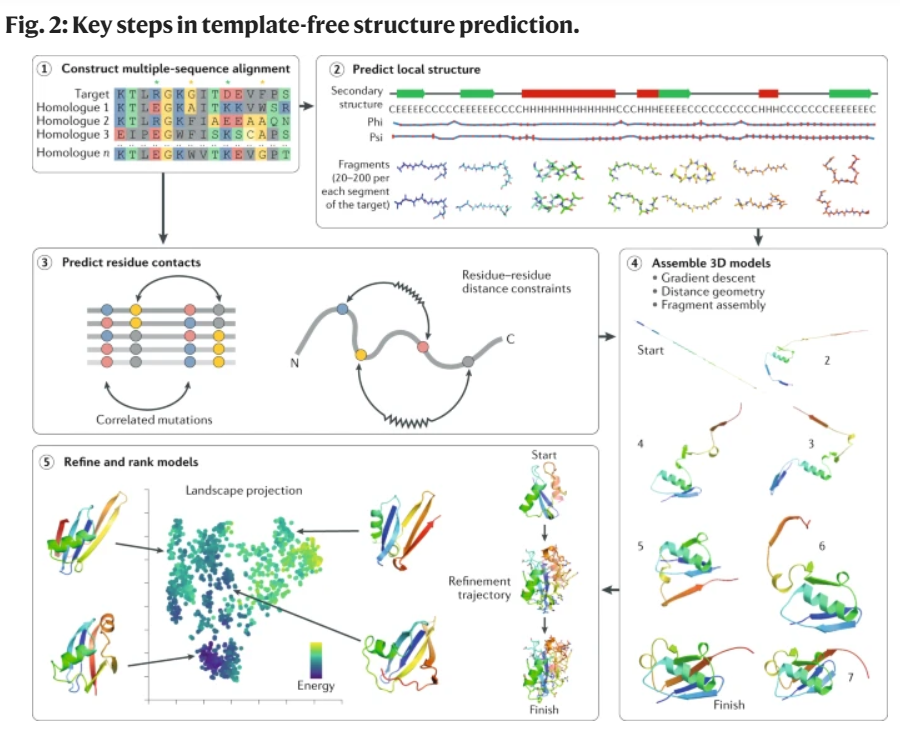
Kernel hyperparameters C and γ are two of them. The penalty for misclassifying a data item is determined by C, and the decision region is represented by γ.

**CHAPTER FIVE: TEMPLATE-BASED MODELLING**

The steps in standard template-based modelling include selection of a suitable structural template; alignment of the target sequence to the template structure; and molecular modelling to account for mutations, insertions and deletions present in the target–template alignment. Closely related templates can be detected by using single-sequence search methods such as BLAST28 to scan the PDB sequences. To detect more distantly related templates, a target sequence profile29,30 built from a multiple-sequence alignment can be used to scan a database of sequence profiles for proteins of known structure by profile–profile comparison31,32 or can be matched to a library of structural templates to assess sequence–structure compatibility33,34. Template selection methods return an initial target–template alignment that can be adjusted manually, often in an iterative manner after model building. Given an alignment to a template, established tools35,36,37 can be used to quickly construct molecular models of the target sequence by performing side-chain optimization only at mutated positions and by rebuilding the backbone around insertions and deletions. For target protein sequences that are only distantly related to proteins of known structure, more sophisticated approaches that rely on multiple templates and perform aggressive backbone conformational sampling may be required37,38,39. Together with available crystal structures, template-based modelling approaches can provide structural information for roughly two-thirds of known protein families40.

**5.1 Template-Free Modelling**

Template-free modelling approaches can be applied to proteins without global structural similarity to a protein in the PDB. Lacking a structural template, these methods require a conformational sampling strategy for generating candidate models, as well as a ranking criterion by which native-like conformations can be selected. The structure prediction process without a template (Fig. 2) typically begins with the construction of a multiple-sequence alignment of the target protein and related sequences. The sequences of the target and its homologues are then used to predict local structural features, such as secondary structure and backbone torsion angles, and non-local features, such as residue–residue contacts or inter-residue distances across the polypeptide chain. These predicted features guide the process of building 3D models of the target protein structure, which are then refined, ranked and compared with one another to select the final predictions.



An accurate multiple-sequence alignment between the target protein and its sequence homologues contains valuable information on the amino acid variation between the homologous sequences, including correlated patterns of sequence changes occurring at different positions (the green and yellow stars highlight pairs of alignment columns displaying amino acid charge and size swapping, respectively) (step 1). The target sequence and the multiple-sequence alignment form the basis for predictions of local backbone structure, including torsion angles (phi and psi predictions are shown, with red error bars indicating uncertainty) and secondary structure (step 2; PSIPRED67 predictions are shown). Libraries of backbone fragments taken from proteins predicted to have similar local structures can also be assembled for use in model building. The multiple-sequence alignment can be used to predict residue pairs likely to be in spatial contact on the basis of observation of correlated mutations in pairs of alignment columns (step 3). These predictions of local structure and residue contacts guide 3D model building with techniques such as gradient-based optimization, distance geometry or fragment assembly (step 4; snapshots from a Rosetta42 fragment assembly trajectory are shown). Initial 3D models are typically built with a reduced representation and a coarse-grained energy function; to better determine near-native predictions, these models are refined with an all-atom energy function and compared with one another to identify clusters of similar low-energy conformations, from which representative models are chosen as the final predictions (step 5; a 2D principal-component projection of the space of refined models is shown, in which each dot represents a single model).

**Related Work**

Protein secondary structure prediction from sequences is then regarded as an intermediate challenge that bridges the gap between the prediction of primary and tertiary structure. To predict protein secondary structure, a variety of supervised and unsupervised approaches have been used in prior studies. The secondary structure of the protein has been predicted using a variety of machine and deep learning algorithms including the support vector machines [23, 20], probability graph models [24, 25], artificial neural network [26, 27, 28, 29], hidden Markov models [30, 31], bidirectional recurrent neural network(BRNN) [32, 33, 34], and so on.

Protein secondary structures have traditionally been classified into three states. But the secondary assignment’s eight aspects were devised in 1983 [21] and are still in use today. To increase the efficiency of protein secondary structure prediction, this research[35] introduces a revolutionary deep learning architecture that uses an integrated synergy of prediction by a residual network, convolutional neural network and bidirectional recurrent neural network. On the benchmark CB513 dataset, their proposed deep network obtained 71.4% accuracy for 8-state prediction and 74% accuracy for ensemble learning.

Even when the target protein’s sequence differs significantly from the reference proteins, nearest-neighbor-based approaches estimate the secondary structure of a target protein based on local sequence similarities to portions of known proteins, often through a sliding window. NNSSP [36] and Preator [37] are the most widely used nearest neighbor prediction servers. The NNSSP server, which uses multiple sequence alignments to incorporate evolutionary information, has a 72.2% accuracy rate. In contrast to multiple sequence alignment, the Predator uses the local pair-wise alignment of the target sequence.

Hidden Markov Models, similar to the nearest neighbor technique, have been used to predict secondary structure. Following the construction of a multiple sequence alignment profile utilizing short segments of comparable sequences with known structure, hidden Markov Models in a structured context are produced and used to predict the structure of the unknown protein [38]. This approach is used by Bystroff et al. in their program HMMSTR, which claims 74.3% accuracy [39].

The neural network method relies on how synaptic connections in brain neurons function, where input is processed at multiple levels before being mapped to the final output. Using a training set of sequences with known structures, neural networks are trained by modifying the values of the weights that affect signals. Typical neural network techniques include Rost and Sander’s [40] PHD program, which claims 72.2% accuracy after being trained on a test set of profiles from different sequences. When larger databases and PSI-BLAST[27] were utilized to build the training set [36], this accuracy was enhanced to 75%.

To predict the secondary structure of the protein, Kathuria et al. [8] utilize a Random Forest classification approach. Breiman developed Random Forest in 2001 as a classification and regression tree (CART) technique that can generate a large number of decision trees from a single set of training data [41]. They only classify alpha (*α*) structure and non-alpha structure in this paper. During preprocessing step, all alpha structure polypeptides have their structure-property set to alpha, whereas all other polypeptide structures are set to non-alpha. Their model is validated using a ROC curve to show that it produces the intended outcomes. They did, however, limit themselves to binary classification in their research.

Support vector machines (SVM) and artificial neural networks (ANN) are used to predict secondary structure in the article [42]. They strive to increase their accuracy by converting the 8 secondary protein states (G, E, I, H, T, S, B, C) into three-state (H, C, E).

Both the primary sequences and the targets are converted to binary forms in these approaches, and prediction is done using a sliding window [43, 44]. Different binary classifiers have varied levels of performance when utilizing an ad hoc sliding window size in NN and SVM techniques [45, 20]. On 126 non-homologous data sets (RS126), Rost et al. [46] used threelayer feed-forward NNs with protein evolutionary information to achieve an accuracy of Q3 = 70.8%. To solve the overfitting problem, Riis et al. [47] proposed using NNs with minor neural networks to predict three states of protein secondary structure. Additionally, they achieved 71.3% accuracy on the RS126 set by utilizing another NN and integrating multiple alignment information as a feature. With a bigger training data set of 681 non-homologous proteins, Chandonia et al. [48] developed a new technique for PSSP and achieved 74.8% accuracy.

The support vector machine (SVM) was proposed by Vapnik et al. [49] as an alternate solution for classification issues that uses frequent profiles and evolutionary information as an encoded input [20]. Other researchers exploited PSI-BLAST PSSM profiles [50] as input vectors and combined the sliding window approach with SVM architecture to achieve a 70 percent accuracy [51]. SVM has steadily been used to PSSP problems since it has occasionally beaten most other learning systems, including NNs [52].

Ghosh et al. [1] used a minimal distance classifier via several window sizes, and they were able to achieve an accuracy of roughly 60%. Ahmed et al. employed a genetic algorithm with a multiple window technique to achieve a Q3 = 68 percent accuracy. Spencer and colleagues[53] have created a protein secondary structure predictor based on the positionspecific scoring matrix BLAST and deep learning network topologies that were optimized and quickly trained on a set of 198 proteins.

The architecture of a Graph Neural Network (GNN) is explained by Zhang et al. [54]. It is impressive to apply GNN in the process of predicting the secondary structure of the protein. GNN is used to predict the secondary structure of a protein as a result of this inspiration. With all of this motivation in mind, the use of a graph neural network in protein secondary structure has been considered.