

## Term Project

# COMPUTATIONAL BIOPHYSICS

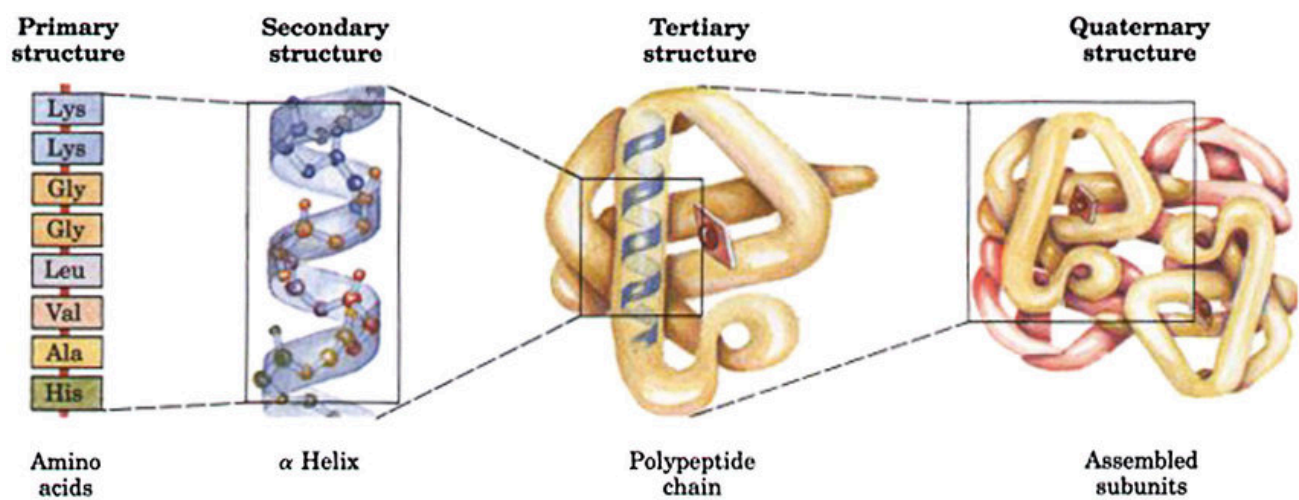
April 15, 2024

Group 1

- Hardik Pravin Soni (20CS30023)

- Nikhil Saraswat (20CS10039)

- Vivek Jaiswal (20CS10077)



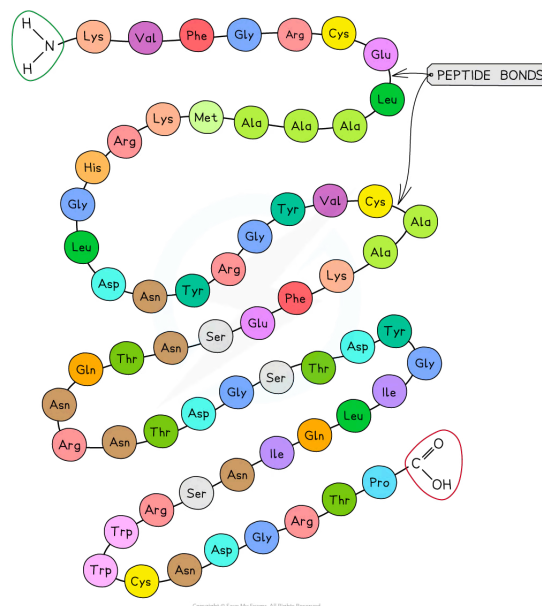
## Protein

A protein is a complex organic molecule made up of amino acids. Proteins play crucial roles in almost all biological processes, such as catalyzing biochemical reactions, providing structure to

---

cells and tissues, and serving as signaling molecules. They are essential for the structure, function, and regulation of the body's tissues and organs.

## Primary Structure



The primary structure of a protein refers to the linear sequence of amino acids linked together by peptide bonds. It represents the specific order in which the amino acids are arranged along the polypeptide chain. This sequence is determined by the genetic code encoded in an organism's DNA. The primary structure of a protein is fundamental as it dictates the folding and ultimately the function of the protein molecule.

---

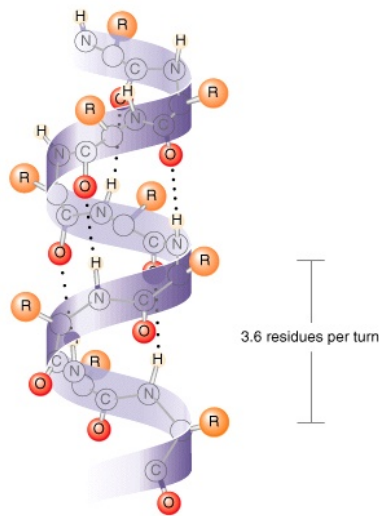
## Secondary Structure

Secondary structure refers to the local folding patterns within a protein's polypeptide chain. These structures are stabilized by hydrogen bonds between the amino acid residues. The two most common types of secondary structures are alpha helices and beta sheets. Additionally, there are turns or loops, which connect secondary structures or reverse the direction of the polypeptide chain.

### Type of secondary structures: -

Here are the three main types of secondary structures:

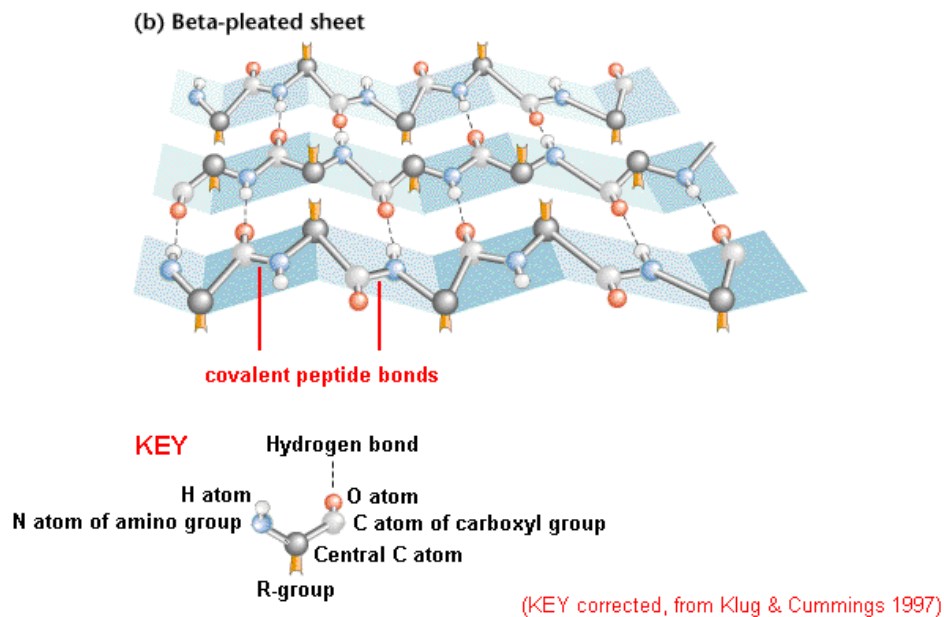
#### Alpha Helix:



- In an alpha helix, the polypeptide chain coils into a right-handed spiral or helix.

- The backbone hydrogen bonds form between the carbonyl oxygen of one amino acid and the amine hydrogen of an amino acid three or four residues down the chain.
- The side chains of the amino acids protrude outward from the helix.
- Alpha helices are common in structural proteins and are often found in regions of proteins that span cellular membranes.

## Beta Sheet:



- In a beta sheet, the polypeptide chain folds back and forth, forming a sheet-like structure.
- The backbone hydrogen bonds form between adjacent segments of the polypeptide chain, which may run in the same (parallel) or opposite (antiparallel) directions.
- The side chains of the amino acids alternate above and below the plane of the sheet.

- 
- Beta sheets are important for providing strength and rigidity to protein structures and are often found in the core of globular proteins.

## Turns or Loops:

### Turns and Loops



- Non-regular secondary structure
- Most proteins are globular
  - Strands and sheets are short
  - Hydrophobic cores require chains to fold back on themselves
  - Turns and loops allow chain reversal

1

- Turns or loops are regions of the polypeptide chain where it changes direction abruptly.
- These regions often connect secondary structure elements or reverse the direction of the polypeptide chain.
- Turns contribute to the overall folding and tertiary structure of proteins, and they can be involved in protein-protein interactions or binding sites.

Additionally, beyond the main three types, there are specific variations and combinations of secondary structures, leading to eight recognized types:

1. C: Loops and irregular elements (corresponding to the blank characters output by DSSP)

- 
2. E:  $\beta$ -strand
  3. H:  $\alpha$ -helix
  4. B:  $\beta$ -bridge
  5. G: 3-helix
  6. I:  $\pi$ -helix
  7. T: Turn
  8. S: Bend


## About Dataset

### Introduction

Protein secondary structure can be calculated based on its atoms' 3D coordinates once the protein's 3D structure is solved using X-ray crystallography or NMR. Commonly, DSSP is the tool used for calculating the secondary structure and assigns one of the following secondary structure types (<https://swift.cmbi.umcn.nl/gv/dssp/index.html>) to every amino acid in a protein:

1. C: Loops and irregular elements (corresponding to the blank characters output by DSSP)
2. E:  $\beta$ -strand
3. H:  $\alpha$ -helix
4. B:  $\beta$ -bridge
5. G: 3-helix
6. I:  $\pi$ -helix
7. T: Turn
8. S: Bend

However, X-ray or NMR is expensive. Ideally, we would like to predict the secondary structure of a protein based on its primary sequence directly, which has had a long history. A review on this topic



---

was published recently, Sixty-five years of the long march in protein secondary structure prediction: the final stretch?.

For the purpose of secondary structure prediction, it is common to simplify the aforementioned eight states (Q8) into three (Q3) by merging (E, B) into E, (H, G, I) into E, and (C, S, T) into C. The current accuracy for three-state (Q3) secondary structure prediction is about ~85% while that for eight-state (Q8) prediction is <70%. The exact number depends on the particular test dataset used.


## Dataset

The main dataset lists peptide sequences and their corresponding secondary structures. It is a transformation of <https://cdn.rcsb.org/etl/kabschSander/ss.txt.gz> downloaded from [RSCB PDB](https://www.rcsb.org/) into a tabular structure. If you download the file at a later time, the number of sequences in it will probably increase.

### Description of columns:

1. **pdb\_id**: the id used to locate its entry on <https://www.rcsb.org/>
2. **chain\_code**: when a protein consists of multiple peptides (chains), the chain code is needed to locate a particular one.
3. **seq**: the sequence of the peptide
4. **sst8**: the eight-state (Q8) secondary structure
5. **sst3**: the three-state (Q3) secondary structure
6. **len**: the length of the peptide
7. **has\_nonstd\_aa**: whether the peptide contains nonstandard amino acids (B, O, U, X, or Z).

### Key steps in the transformation:

- Both Q3 and Q8 secondary structure sequences are listed.
- 

- 
- All nonstandard amino acids, which includes B, O, U, X, and Z, (see here for their meanings) are masked with "\*" character.
  - An additional column (has\_nonstd\_aa) is added to indicate whether the protein sequence contains nonstandard amino acids.
  - A subset of the sequences with low sequence identity and high resolution, ready for training, is also provided

For details of curation, please see [Link](#).

A subset (9079 sequences) based on sequences culled by PISCES with more strict quality control is also provided. This dataset is considered ready for training models.

The culled subset generated on 2018-05-31 with cutoffs of 25%, 2Å, and 0.25 for sequence identity, resolution and R-factor respectively, is used. The URL to the original culled list is [http://dunbrack.fccc.edu/Guoli/culledpdb\\_hh/cullpdb\\_pc25\\_res2.0\\_R0.25\\_d180531\\_chains9099.gz](http://dunbrack.fccc.edu/Guoli/culledpdb_hh/cullpdb_pc25_res2.0_R0.25_d180531_chains9099.gz), but it may not be permanently available. This dataset contains more columns from [cullpdb\\_pc25\\_res2.0\\_R0.25\\_d180531\\_chains9099.gz](#) with self-explanatory names.


For more about PISCES, please see

<https://academic.oup.com/bioinformatics/article/19/12/1589/258419>.

## Acknowledgements

The peptide sequence and secondary structure are downloaded from [Link](#).

The culled subset is downloaded from [Link](#).





# Snapshot of Datasets

2018-06-06-ss.cleaned.csv (315.17 MB)

/kaggle/input/protein-secondary-structure/2018-06-06-ss.cleaned.csv

Copy

Δ pdb_id	Δ chain_code	Δ seq	Δ sst8	Δ sst3	# len	✓ has_nonst...
1A3θ	C	EDL	CBC	CEC	3	False
1Bθ5	B	KCK	CBC	CEC	3	False
1BθH	B	KAK	CBC	CEC	3	False
1B1H	B	KFK	CBC	CEC	3	False
1B2H	B	KAK	CBC	CEC	3	False
1B32	B	KMK	CBC	CEC	3	False
1B3F	B	KHK	CBC	CEC	3	False
1B3G	B	KIK	CBC	CEC	3	False

2018-06-06-pdb-intersect-pisces.csv (6.98 MB)

/kaggle/input/protein-secondary-structure/2018-06-06-pdb-intersect-pisces.csv

Copy

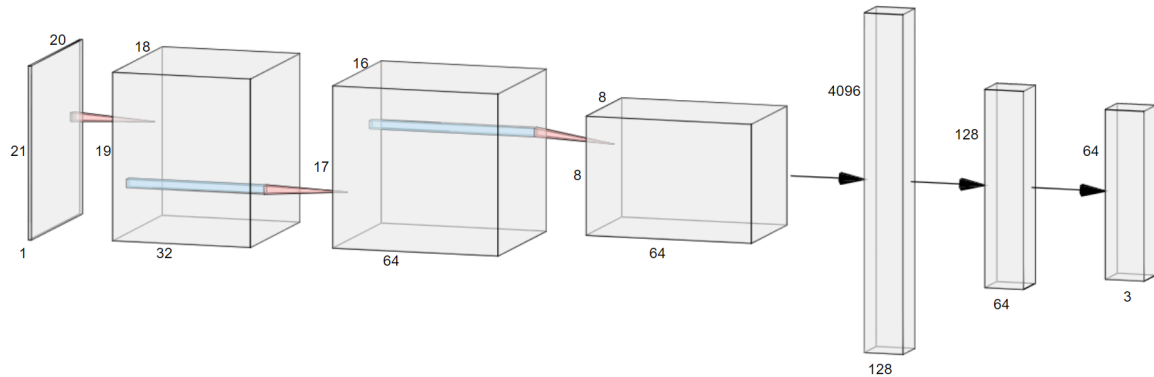
11 of 11 columns

Δ pdb_id	Δ chain_code	Δ seq	Δ sst8	Δ sst3	# len	✓ has_nonst...	Δ Exptl.	# resolution	# R-factor	# FreeRvalue
1FV1	F	NPVVHFFKNIVTPRT PPPSQ	CCCCC8CCCCCCCC CCCC	CCCCCECCCCCCCC CCCC	2θ	False	XRAY	1.9	θ.23	θ.27
1LM8	H	DLGLEMLAPYIPMD DFQLR	CCCCCCCC8CCSCC CEEC	CCCCCCCCCECCCC CEEC	2θ	False	XRAY	1.85	θ.2	θ.24
1θB6	A	EEDPDLKAAIQESLR EAEEA	CCCHHHHHHHHHHH HHHTC	CCCHHHHHHHHHHH HHHCC	2θ	False	XRAY	1.45	θ.19	θ.22
1QOW	D	CTFTLPGGGGVCTLT SECI*	CCTTSC7CSSTTSS TTCCC	CCCCCCCCCCCCCCC CCCC	2θ	True	XRAY	1.θ6	θ.14	1.θ
1RDQ	I	TTYADFIASGRTGRR NAIHD	CHHHHHHTSSCSSCC CCEEC	CHHHHHHCCCCCCCC CCEEC	2θ	False	XRAY	1.26	θ.13	θ.16
1T6θ	B	QDSRRSADALLRLQA	CHHHHHHHHHHHHH	CHHHHHHHHHHHHH	2θ	False	XRAY	2.θ	θ.23	θ.28

---

# Model Architectures:

## Model Architecture for RNN Model



The provided neural network architecture can be described as follows:

**Input Layer:** The input to the network is a 2D image tensor with a single channel (grayscale image).

### Convolutional Layers:

- Two convolutional layers are employed, each followed by batch normalization and ReLU activation.
- The first convolutional layer (conv1) has 1 input channel and 32 output channels, followed by batch normalization (bn\_2d\_a).
- The second convolutional layer (conv2) has 32 input channels and 64 output channels, followed by batch normalization (bn\_2d\_b).

### Pooling Layer:

- 
- Max pooling with a kernel size of 2x2 and a stride of 2 is applied to reduce spatial dimensions.

#### **Dropout Layer:**

- Two dropout layers are incorporated for regularization:
- The first dropout (dropout1) with a rate of 0.25 is applied after the pooling layer.
- The second dropout (dropout2) with a rate of 0.5 is applied before the final fully connected layer.

#### **Recurrent Layer:**

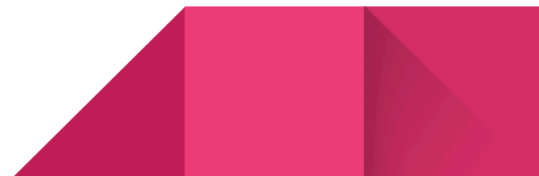
- An RNN layer (rnn) is utilized with an input size of 8, a hidden size of 64, and 1 layer.
- The RNN operates in a batch-first manner, taking sequences of input data.

#### **Fully Connected Layer:**

- After flattening the output of the RNN, it is passed through fully connected layers.
- The first fully connected layer (fc1) has 128 neurons, followed by batch normalization (bn\_1d\_a) and ReLU activation.
- The second fully connected layer (fc2) has 64 neurons, followed by batch normalization (bn\_1d\_b) and ReLU activation.
- The final fully connected layer (fc3) produces the output logits with the desired output size.

#### **Output Layer:**

The output of the final fully connected layer is passed through a log softmax activation function along dimension 1, producing the output probabilities.



---

## For LSTM Q3:

The provided function creates a Long Short-Term Memory (LSTM) based model with optional attention mechanism. Here's a breakdown of the architecture:

### **Input Layer:**

The input layer takes sequences of integer indices representing words in sentences. The shape of the input tensor is (None,), where None signifies variable sequence length.

### **Embedding Layer:**

An embedding layer converts input word indices into dense vectors of fixed size. The embedding size is 128 in this case.


### **Bidirectional LSTM Layers:**

- One or more bidirectional LSTM layers are stacked.
- Each LSTM layer has 64 units and returns sequences (i.e., it operates on each timestep).
- A dropout layer is applied after each LSTM layer to prevent overfitting. The dropout rate is specified by the dropout\_rate parameter.

### **Attention Mechanism:**

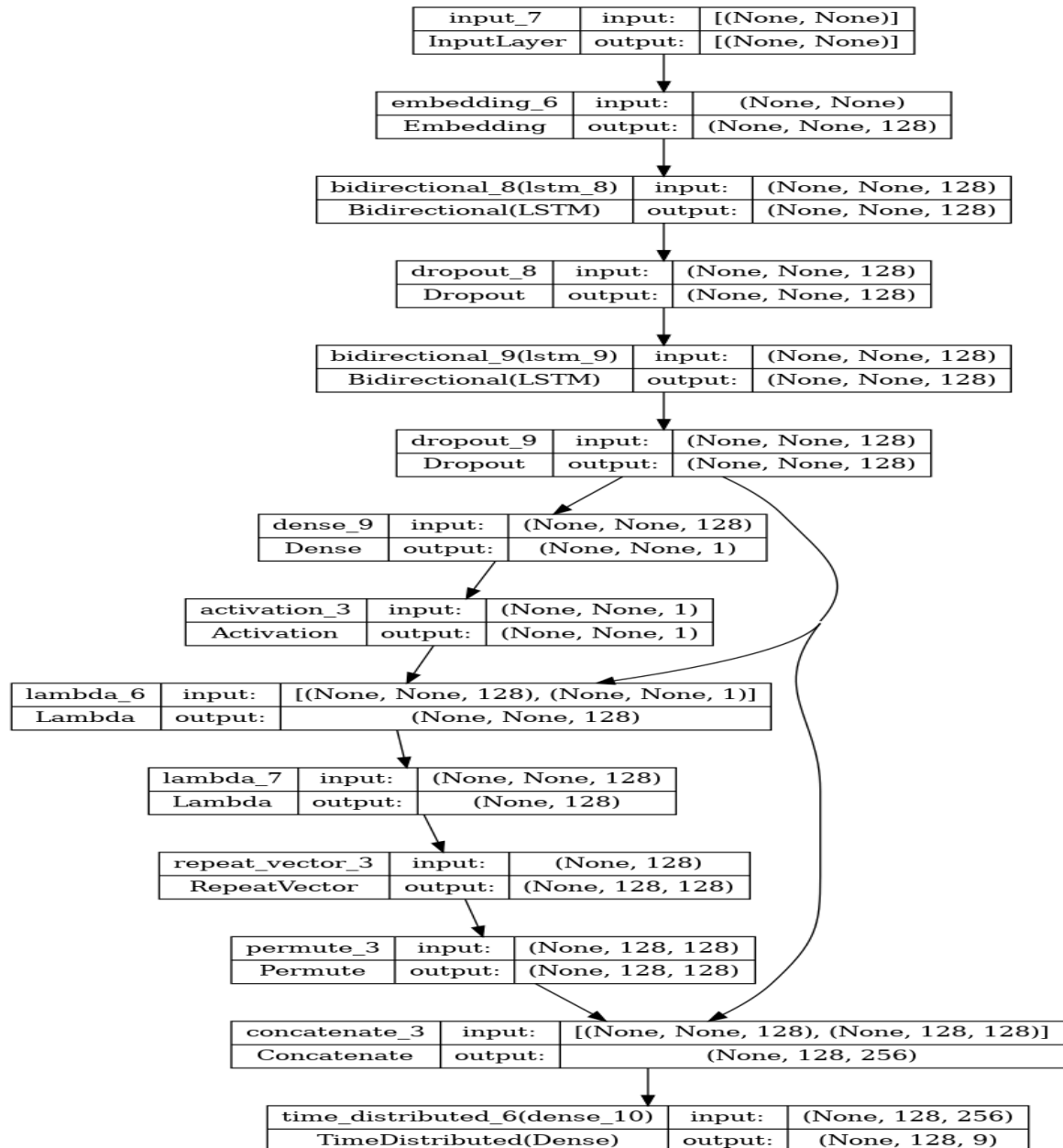
- If attention=True, an attention mechanism is added after the LSTM layers.
- The attention mechanism calculates attention weights using a neural network with a single dense layer with a tanh activation function followed by a softmax activation function.
- The attention weights are applied to the LSTM output sequences to obtain a weighted sum, which captures the context or importance of different parts of the input sequence.
- The resulting attention vector is concatenated with the LSTM output sequences.

### **TimeDistributed Dense Layer:**

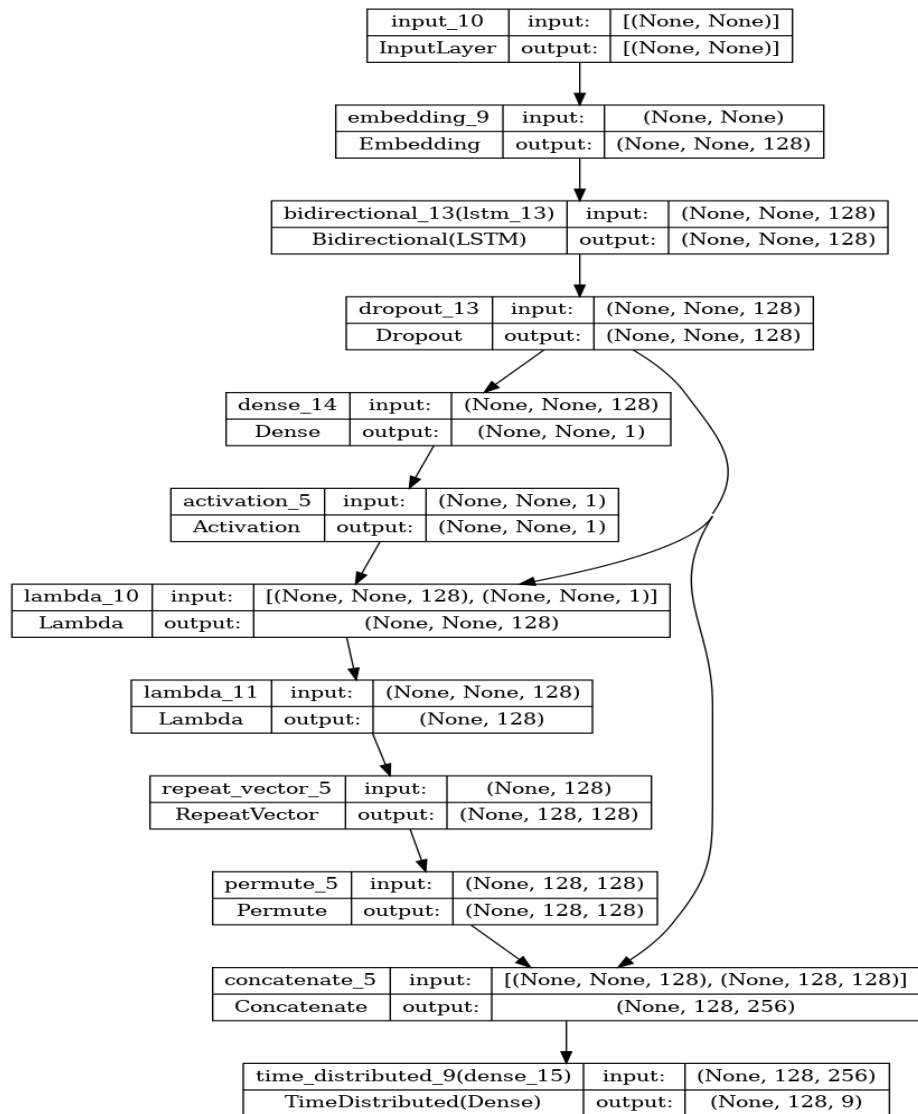
- A time-distributed dense layer is applied to each timestep of the output sequence from the LSTM layers.
  - The dense layer has num\_tags units with softmax activation, producing a probability distribution over the tags for each word in the input sequence.
- 

## Output Layer:

The output of the time-distributed dense layer is the final output of the model, representing the predicted tags for each word in the input sequence.



## For LSTM Q8:



---

## For Transformer:

The provided function creates a Transformer model for sequence labeling tasks. Here's a breakdown of the architecture:

### Input Layer

The input layer (inputs) takes sequences of integer indices representing words in sentences. The shape of the input tensor is (None,), where None signifies variable sequence length.

### Embedding Layer

- An embedding layer converts input word indices into dense vectors of dimension `d_model`.
- The embedding layer has a vocabulary size of `num_words` and embedding dimension `d_model`.

### Positional Encoding Layer:

- Positional encoding is added to the embedded input sequences to provide positional information to the model.
- The positional encoding is calculated using a positional encoding function (`positional_encoding`) based on the position of each token in the sequence.
- The positional encoding is added element-wise to the embedded input sequences.

### Transformer Layers

The Transformer model consists of `num_layers` identical layers. Each layer includes two sub-layers:

- **Multi-head Attention Layer:** Computes self-attention over the input sequence. The attention mechanism allows the model to weigh the importance of different words in the input sequence when making predictions.



---

The output of the attention layer is combined with the input using residual connections and layer normalization.

- **Position-wise Feedforward Layer:** Applies a fully connected feedforward network to each position separately and identically.

The output of the feedforward layer is combined with the output of the attention layer using residual connections and layer normalization.

### **Output Layer**

The output of the Transformer layers is passed through a time-distributed dense layer (`TimeDistributed(Dense)`) with softmax activation. This layer produces a probability distribution over the tags for each word in the input sequence.

### **Model Creation**

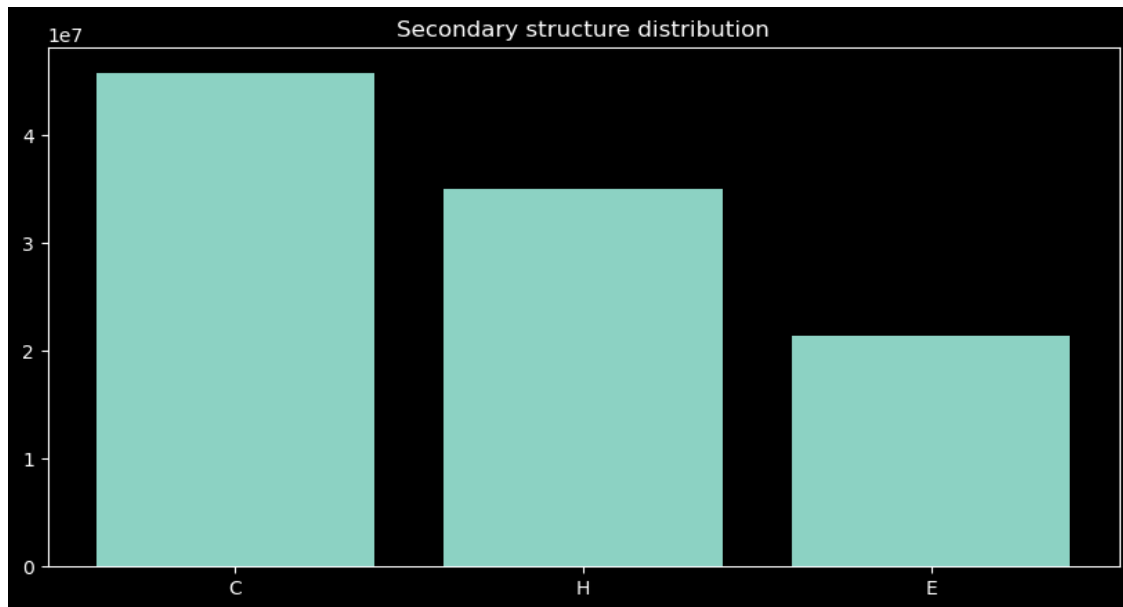
The Keras Model is created with the input layer (inputs) and the output layer (outputs). This architecture is suitable for sequence labeling tasks, such as part-of-speech tagging or named entity recognition. The Transformer model excels at capturing long-range dependencies in sequences and has been widely adopted in natural language processing tasks.



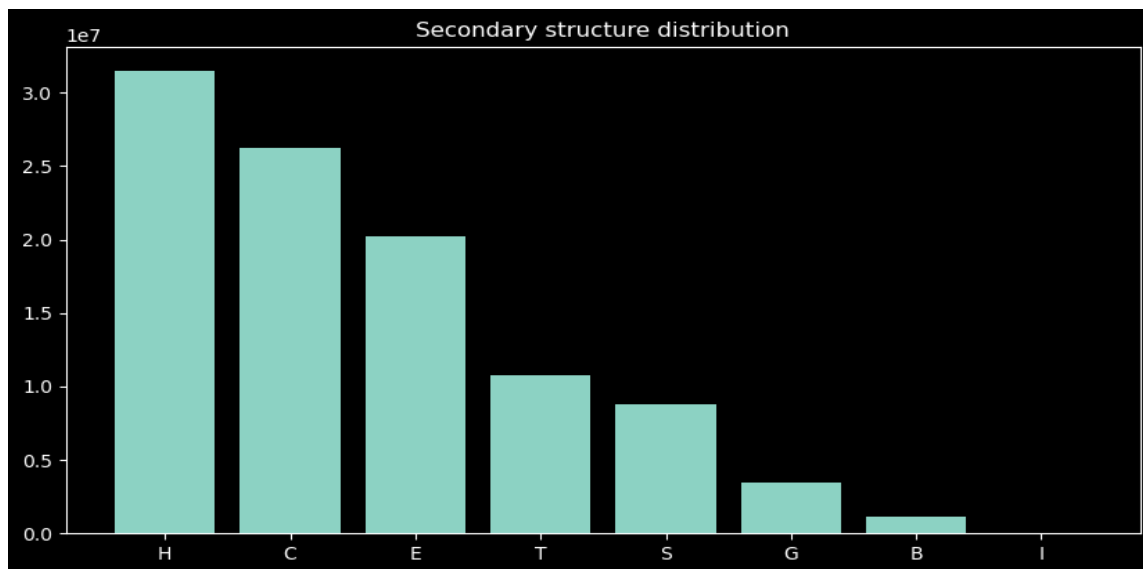


---

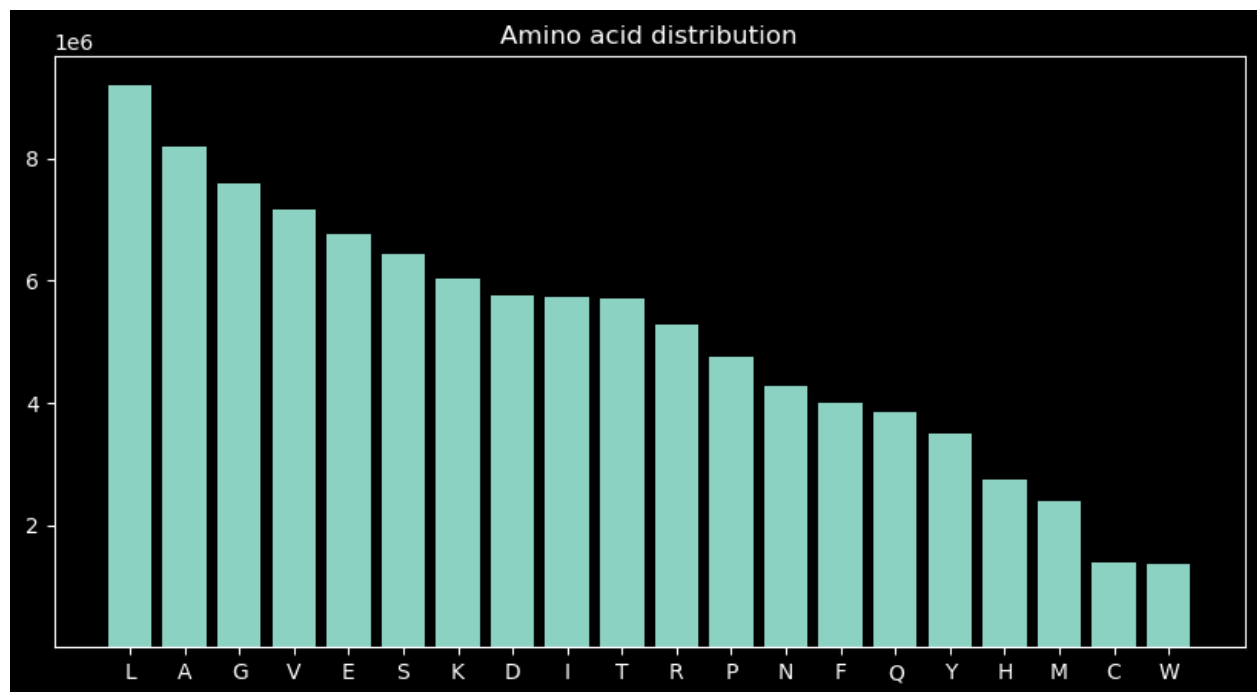
## Plots and Results:



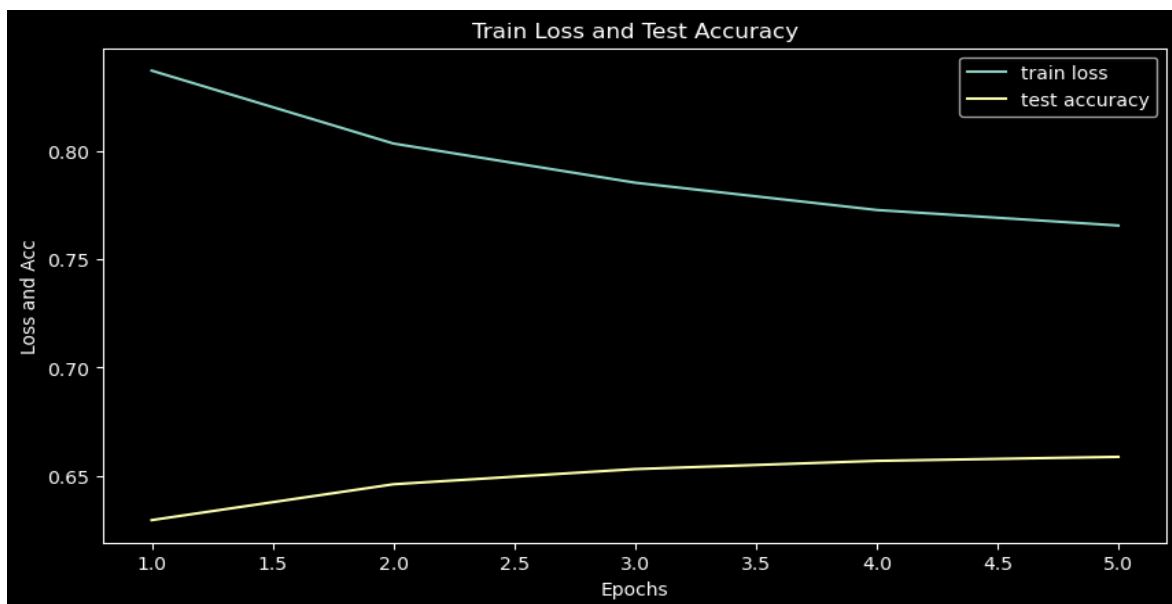
**Figure 1:** Secondary Structure Distribution for **sst3**



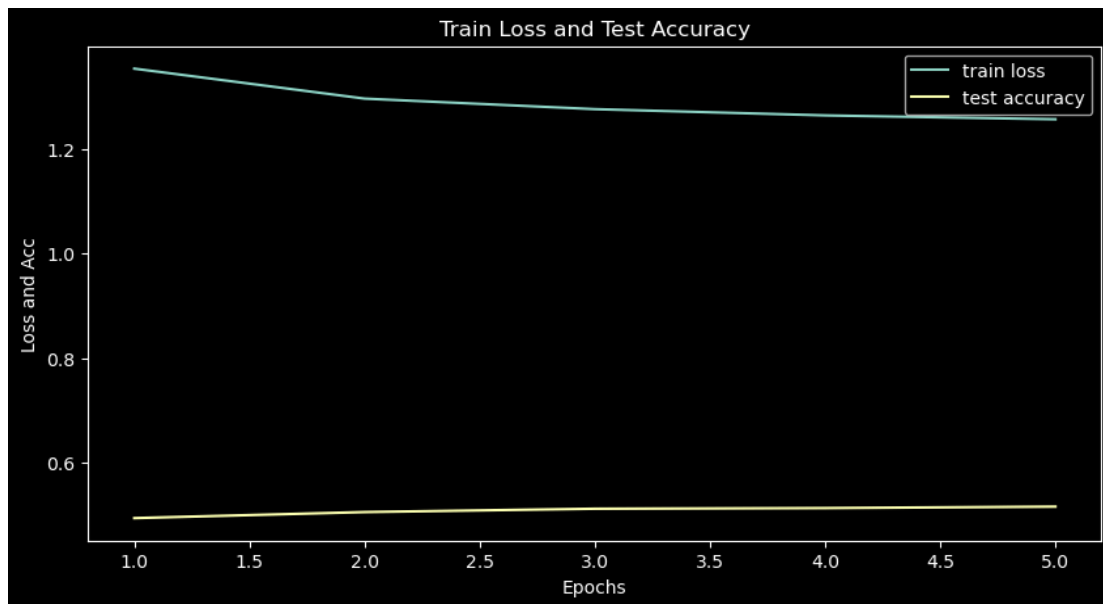
**Figure 2:** Secondary Structure Distribution for `sst8`



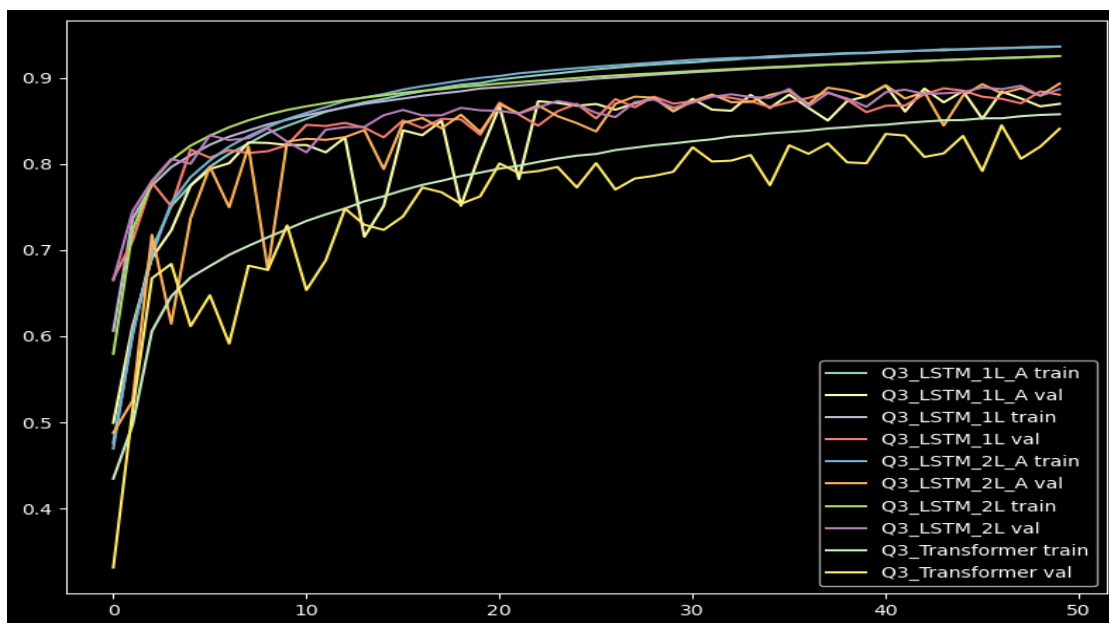
**Figure 3:** Distribution of Amino Acids in the Dataset



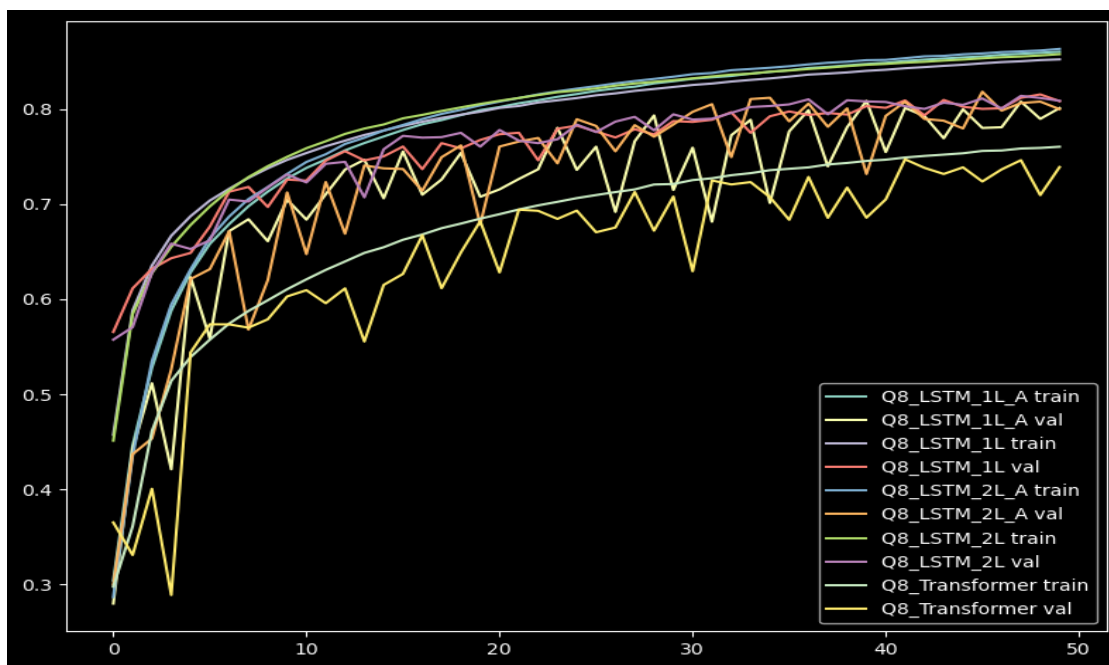
**Figure 4:** Training Loss and Test Accuracy for RNN for Q3 Type



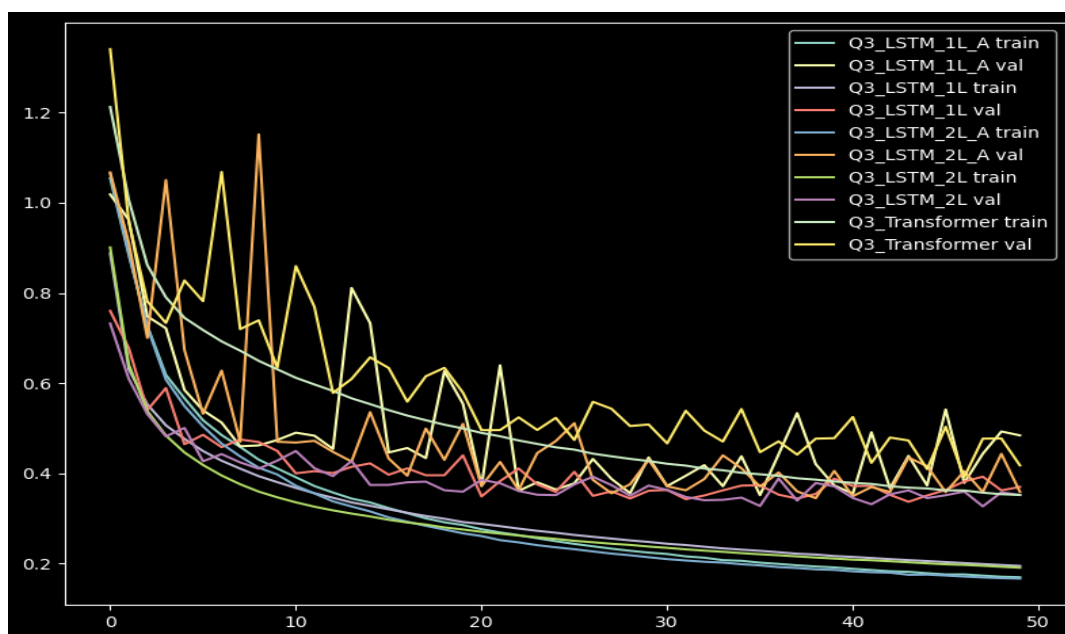
**Figure 5:** Training Loss and Test Accuracy for RNN for Q8 Type



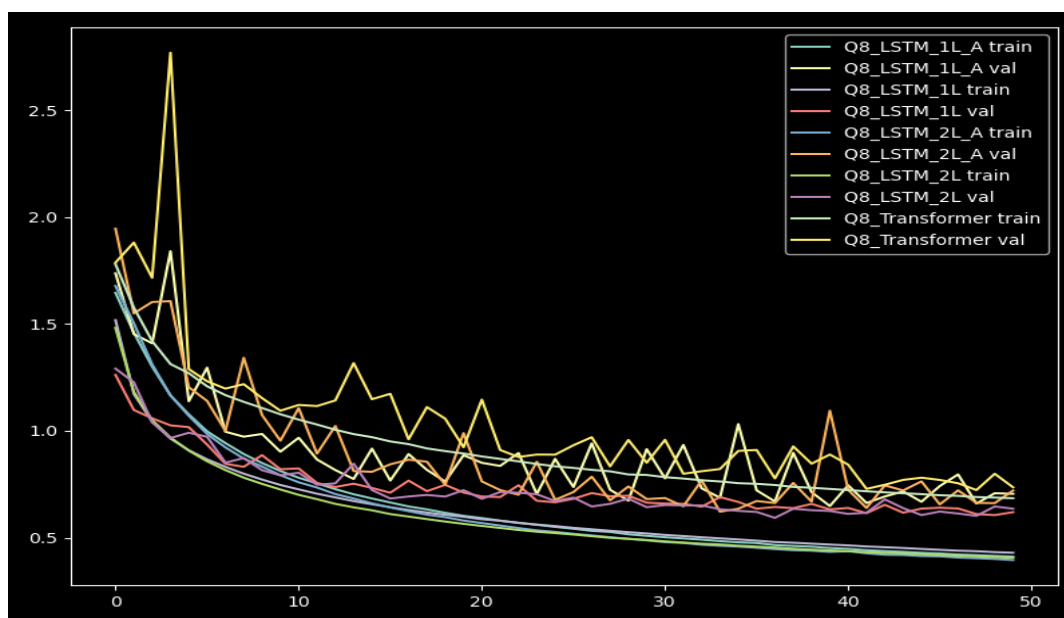
**Figure 6:** Accuracy Plot for Q3 Type for LSTM and Transformer Models



**Figure 7:** Accuracy Plot for Q8 Type for LSTM and Transformer Models



**Figure 8:** Loss Values for Q3 Type for LSTM and Transformer Models



**Figure 9:** Loss Values for Q8 Type for LSTM and Transformer Models

## Observation:

### Summary of Numerical Results Obtained:

- All amino acid sequences are merged.
- 20 amino acid and their **sst3** [3 secondary structures (C, E, and H)] or **sst8** [8 secondary structures] are used.
- Each amino acid in the window is encoded using Tokenizer function from `keras.preprocessing.text` which does word-based tokenization.
- Tried models are RNN, LSTM and Transformer models with their various configurations mentioned in the but found little affect on accuracy.

- 
- The performance of models generally remains same for number of bidirectional layers and it increase for the case of two layers when attention type mechanisms are used.

## Benchmarks:


### RNN Type Models:-

- For **Q3** Feature the Test set has an **Average loss of 0.7771** and **Accuracy: 290851/441446 (66%)**. The Training Set Provides an **Average Loss: 0.693066**.
- For **Q8** Feature the Test set has an **Average loss: 1.3133** and **Accuracy: 227842/441446 (52%)**. The Training Set Provides an **Average Loss: 1.053971**.

### LSTM Type Models:-

- The Best Models for Q3 Type of Feature for LSTM Models **Q3\_LSTM\_2L\_A:-**  
**loss: 0.1665 - acc: 0.9363 - val\_loss: 0.3588 - val\_acc: 0.8934.**
- The Best Models for Q8 Type of Feature for LSTM Models **Q8\_LSTM\_2L:-**  
**loss: 0.4059 - acc: 0.8579 - val\_loss: 0.6357 - val\_acc: 0.8089.**

### Transformer Type Models:-

- For Q3 Type of Feature for Transformer Models **loss: 0.3520 - acc: 0.8577 - val\_loss: 0.4175 - val\_acc: 0.8409.**
  - For Q8 Type of Feature for Transformer Models **loss: 0.6847 - acc: 0.7605 - val\_loss: 0.7358 - val\_acc: 0.7392.**
- 

---

## Dependencies:

- PyTorch
- Pandas
- Numpy
- Matplotlib
- Seaborn
- scikit-learn
- torchsummary
- Tested on Python 3.8.3 x64

## References:

---

1. Baldi, Pierre, Søren Brunak, Paolo Frasconi, Gianluca Pollastri and Giovanni Soda. "Bidirectional Dynamics for Protein Secondary Structure Prediction." [Sequence Learning](#) (2001).
2. Chen, J. and Chaudhari, N. S.. "Protein Secondary Structure Prediction with bidirectional LSTM networks." [Paper](#) presented at the meeting of the Post-Conference Workshop on Computational Intelligence Approaches for the Analysis of Bio-data (CI-BIO), Montreal, Canada, 2005.
3. Sepp Hochreiter, Martin Heusel, Klaus Obermayer; Fast model-based protein homology detection without alignment, [Bioinformatics](#), Volume 23, Issue 14, 15 July 2007, Pages 1728–1736,