Grenze International Journal of Engineering and Technology, June Issue

Extensive Dataset Driven Hybrid Neural Networks for Protein Secondary Structure Prediction

Saket Sontakke[1], Akshat Srivastava[2], Aditya Kshirsagar[3] and Dr. Akshita Chanchlani[4]
[1-4]Department of Computer Science and Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune, India
Email: sontakkesaket9@gmail.com, {akshatsrivastava1975, adityakshirsagar2, akshita.s.chanchlani}@gmail.com

***Abstract*—Protein secondary structure prediction (PSSP) is one of the oldest and most challenging problems in computational biology. Recently, modern machine learning and deep learning algorithms have been utilized to develop techniques for predicting the secondary structures of proteins. The proposed model was trained on an extensive dataset of 128,644 unique protein sequences, sourced from the Dictionary of Protein Secondary Structure (DSSP) [1][2]. The dataset was split into 70% for training, 10% for validation, and 20% for testing. Due to computational constraints, the maximum sequence length was limited to 256. The architecture includes an embedding layer to capture dense vector representations of amino acid sequences, multiple convolutional layers to identify spatial dependencies, a BiLSTM layer for capturing long-term dependencies, and a BiGRU layer for short-term dependencies, all processed through time-distributed dense layers and an output layer. Techniques like dropout and L2 regularization were used to avoid overfitting and improve generalization. The large scale and diversity of the dataset promoted us to design a slightly more complex architecture to adequately learn the intricate patterns in protein sequences. A custom metric – masked accuracy was defined to avoid inclusion of padding in accuracy calculations. This approach achieved a Q3 accuracy of 86.30%, showing that hybrid architectures are effective for PSSP.**

***Index Terms*— Protein Secondary Structure Prediction (PSSP), Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Gated Recurrent Unit (BiGRU), Machine Learning, Deep Learning, DSSP.**

I. INTRODUCTION

Amino acids [3] are building blocks of protein which consist of an amino group (-NH2), a carboxyl group (-COOH) and a side chain (R). Amino acids link through a condensation reaction, where a hydroxyl group (—OH) is removed from one amino acid's carboxyl group and a hydrogen atom (H) from another's amino group, forming water and creating a peptide bond between the two. A chain of amino acids linked by peptide bonds is called a peptide. Long chains of amino acids are referred to as polypeptides and hence proteins are polypeptides made up of many amino acid residues connected by peptide bonds. There are 20 standard or common amino acids as depicted in TABLE I that occur naturally in proteins. Linear sequences of these amino acids fold into functional proteins. Proteins are characterized by four structural levels:

- Primary Structure: The sequence of amino acids.
- Secondary Structure: Local folding into α-helices (H), β-sheets (E), and coils/loops (C).
- Tertiary Structure: The 3D arrangement of a single polypeptide.

Grenze ID: 01.GIJET.11.2.197

- Quaternary Structure: Assembly of multiple polypeptide subunits.

Traditionally, these protein structures are experimentally determined using scientific methods such as X-ray crystallography, Nuclear Magnetic Resonance (NMR) spectroscopy, Cryo-electron microscopy (Cryo-EM). X-ray crystallography is the most common technique, offering high-resolution structures by analyzing the diffraction pattern of X-rays passed through crystallized proteins. NMR spectroscopy allows for the determination of protein structures in solution, which is useful for small proteins and provides insights into protein dynamics. Cryo-EM is a rapidly advancing method that enables the visualization of large protein complexes and membrane proteins without the need for crystallization.

TABLE I. 20 STANDARD AMINO ACIDS AND THEIR CORRESPONDING 1- LETTER ABBREVIATION

Amino acid	Abbreviation	Amino acid	Abbreviation
Alanine	A	Leucine	L
Arginine	R	Lysine	K
Asparagine	N	Methionine	M
Aspartic acid	D	Phenylalanine	F
Cysteine	C	Proline	P
Glutamine	Q	Serine	S
Glutamic acid	E	Threonine	T
Glycine	G	Tryptophan	W
Histidine	H	Tyrosine	Y
Isoleucine	I	Valine	V

Each experimental method for determining protein structures has its limitations. X-ray crystallography, while providing high-resolution structures, requires protein crystallization, which is time-consuming and challenging for some proteins. NMR spectroscopy is limited to small proteins, is time-intensive, and requires large amounts of pure protein. Cryo-EM, though powerful for large complexes, offers lower resolution for small proteins and demands costly equipment and specialized expertise. Primitive methods for protein secondary structure prediction include – Chou-Fasman method [4], GOR method [5], Lim method [6]. The comparisons in TABLE II highlights the key comparisons among these methods.

TABLE II. COMPARISON OF VARIOUS PRIMITIVE METHODS FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Characteristics	Method		
	Chou-Fasman	GOR Method	Lim Method
Year	1974	1978	1974
Approach	Propensity-based, sliding window	Information theory-based, considers neighbouring residues	Empirical rules, emphasis on hydrogen bonding
Accuracy	~50-60%	~60-65%	~50-60%
Strengths	Simple, uses empirical data	Incorporates some context, better accuracy than Chou-Fasman	Simple, rule-based
Weaknesses	Ignores long-range interactions, low accuracy	Still limited by local residue interactions	Low accuracy, ignores complex interactions

These primitive methods, have significant limitations, such as low accuracy, reliance on simple rules or local interactions, and failure to account for complex long-range dependencies in protein structures. Modern Machine Learning (ML) and Deep Learning (DL) algorithms overcome these challenges by leveraging large datasets, advanced architectures, and the ability to model intricate patterns, providing significantly higher accuracy and reliability in protein structure prediction.

Section II provides an overview of related work in the field of protein secondary structure prediction, highlighting the various methodologies and approaches used in recent years. Section III outlines the process of data collection and preparation used to train the proposed model. Section IV contains the specifics of the model's training parameters, callbacks, architecture, and a custom-defined function for accuracy calculation. Section V summarizes the results using various performance metrics. Finally, Section VI offers the conclusion of the study.

II. RELATED WORK

The prediction of protein secondary structure has significantly advanced through the application of deep learning methodologies. Recent studies have employed a range of approaches, utilizing various datasets and achieving notable Q3 accuracies. For instance, Dong et al. [7] proposed SERT-StructNet, a multi-factor hybrid deep learning model designed to enhance prediction accuracy. They evaluated their model on standard datasets,

achieving a Q3 accuracy of 86.5%. Bidirectional architectures such as BiLSTM and BiGRU have also been extensively explored for capturing long-range dependencies. Bongirwar and Mokhade [8] developed a hybrid BiLSTM and BiGRU architecture, reporting a Q3 accuracy of 84.7% on the CB513 dataset. Yang and Chen [9] utilized a hybrid model incorporating convolutional blocks with GRU units, achieving a Q3 accuracy of 85.2% on the same dataset.

Multi-network architectures have demonstrated particular promise. Mohamed et al. [10] introduced Multi-S3P, a deep learning model combining convolutional neural networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM), and self-attention mechanisms. They trained and tested their model on the CB513 and CASP datasets, achieving Q3 accuracies of 85.9% and 86.1%, respectively. Ema et al. [11] explored the integration of CNNs with traditional machine learning algorithms, including Support Vector Machines, Naive Bayes, and Random Forest, to enhance prediction performance, reporting a Q3 accuracy of 83.5% on the CB513 dataset. Similarly, Sutanto et al. [14] demonstrated the effectiveness of combining CNNs with Support Vector Machines in a hybrid model, achieving a Q3 accuracy of 82.7% on the same dataset.

Generative adversarial networks (GANs) have also emerged as a promising tool. Jin et al. [12] developed a model incorporating channel attention and multiscale convolution modules, achieving a Q3 accuracy of 84.9% on the CB513 dataset. Another notable approach, MLPRNN, introduced by Lyu et al. [13], combines a bidirectional gated recurrent unit (BiGRU) with two multi-layer perceptron (MLP) blocks, highlighting the efficacy of integrating recurrent and feedforward networks. They evaluated their model on the CB513 and CASP datasets, achieving Q3 accuracies of 85.5% and 85.8%, respectively. Further innovations include partitioning and semi-random subspace methods (PSRSM), as applied by Ma et al. [15], which leverage data partitioning techniques to enhance model performance. They reported Q3 accuracies of 86.38% on the 25PDB dataset and 84.53% on the CB513 dataset, demonstrating the effectiveness of their approach.

Despite significant advancements in protein secondary structure prediction (PSSP), a critical gap remains in existing methods, as they often fail to utilize large, diverse datasets that cover a wide range of protein sequences. This paper aims to fill that gap by using an extensive dataset with 128,644 protein entries. The model is trained on 90,050 entries and rigorously tested on 25,730 entries. This level of scale is often missing in many existing approaches.

III. Data Preparation

Preparing data is a crucial first step in developing any ML or DL model. Gathering large, authentic, and reliable datasets was a significant challenge. DSSP is a database of secondary structure assignments for all protein entries in the Protein Data Bank (PDB). It is also the name of the program that calculates these assignments from PDB entries. DSSP provides precalculated files for each PDB entry, as well as an application that generates these files. Additionally, a web server is available to obtain individual DSSP files using PDB IDs. An automated Python script was developed using a sample REST API from the DSSP website to collect data in CSV format.

DSSP assigns secondary structure to eight states: bridge (B), β-sheet (E), 310-helix (G), α-helix (H), π-helix (I), Bend (S), Turn (T), and other residues (L). These eight states are often simplified into three states—Helix (H), Sheet (E), and Coil (C)—as outlined in TABLE III.

TABLE III. Assignments of 8-State Structures to their Corresponding 3-State Structures

8-State	3-State
S, T, I, C	C
E, B	E
H, G	H

A. Steps for DSSP Data Processing

- Submit Job: A job is submitted to the server with a PDB ID, and a job ID is returned.
- Check Job Status: The status is polled until it becomes SUCCESS. If marked as FAILURE or
- REVOKED, the process stops.
- Retrieve Results: Once successful, DSSP data is retrieved from the server.

B. Parsing DSSP Content

- Extract Metadata: The PDB ID is extracted from the header line.
- Extract Sequences:
  - The amino acid sequence and DSSP 8-state structure are obtained from the residue table.
  - A DSSP 3-state structure is generated by mapping 8-state to 3-state using predefined rules.

- Output: The processed data, including PDB ID, amino acid sequence, and DSSP structures, is compiled into a CSV file. The dataset contains three key columns: input (amino acid sequences), dssp3 (3-state secondary structure), and dssp8 (8-state secondary structure). While the inclusion of dssp8 is beyond the current scope, it has been considered for potential future exploration.

C. Data Validation

- The sequences are checked for validity by ensuring they only contain valid amino acids - A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y (TABLE I).
- Invalid rows, duplicates and unknown amino acids represented by letter X were dropped from the dataset.

IV. PROPOSED METHOD

Figure 1 shows the architecture of the hybrid neural network for protein secondary structure prediction. It integrates CNN, BiLSTM, and BiGRU layers for sequence processing and classification.

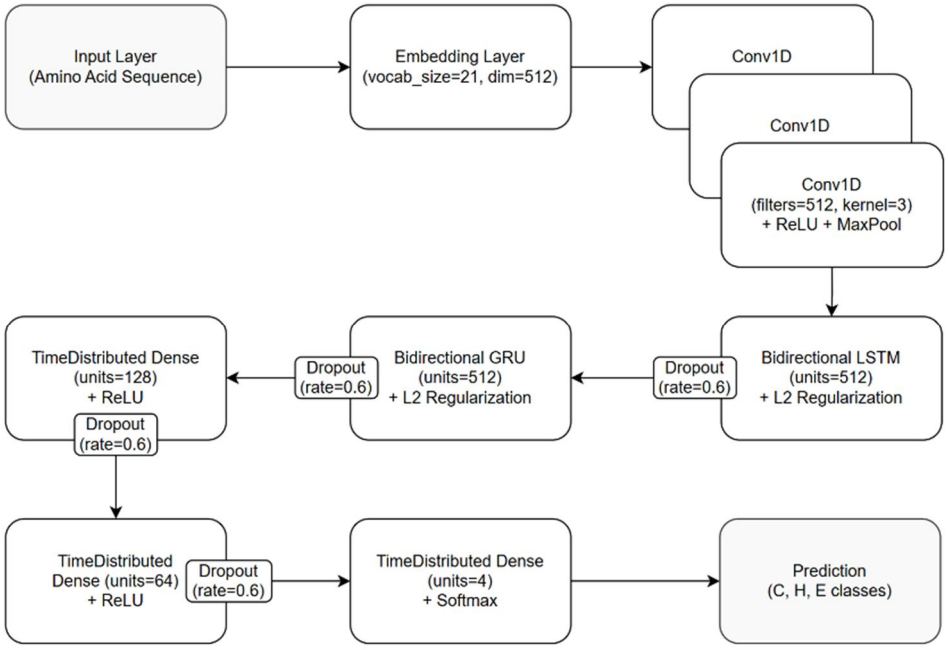

Figure 1. Neural Network Architecture

A. Amino Acid and Secondary Structure Mapping

To prepare sequences for the neural network, amino acids and secondary structure elements are mapped to numerical representations:
Amino Acid Mapping:
{'A': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8, 'K': 9, 'L': 10, 'M': 11, 'N': 12, 'P': 13, 'Q': 14, 'R': 15, 'S': 16, 'T': 17, 'V': 18, 'W': 19, 'Y': 20}
Secondary Structure Mapping:
{'C': 1, 'H': 2, 'E': 3}

B. Padding and Encoding Sequences

To ensure uniform input length for the neural network, sequences are padded to a fixed length of 256 using zeros. After padding, sequences are processed as follows:
Integer Encoding: Each amino acid or secondary structure element is replaced by its corresponding integer based on the mapping.
One-Hot Encoding: Integer labels are converted into binary vectors for multi-class classification. Secondary structure elements (C, H, E) are represented as binary vectors of length 3, with a 1 indicating the presence of a specific class.

Encoded Amino Acid and One-Hot Encoded Secondary Structure Labels

	0	1	2	3	4	5	6	7	8	9	...	246	247	248	249	250	251	252	253	254	255
0	11	18	10	16	4	6	4	19	14	10	...	0	0	0	0	0	0	0	0	0	0
1	11	12	8	5	4	11	10	15	8	3	...	0	0	0	0	0	0	0	0	0	0
2	11	18	10	16	4	6	4	19	14	10	...	0	0	0	0	0	0	0	0	0	0
3	11	12	8	5	4	11	10	15	8	3	...	0	0	0	0	0	0	0	0	0	0
4	11	12	8	5	4	11	10	15	8	3	...	9	13	18	20	3	16	10	3	1	18
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
128639	1	17	10	9	3	14	10	8	7	12	...	18	1	3	10	1	4	16	8	11	9
128640	1	12	14	1	16	18	18	1	12	14	...	0	0	0	0	0	0	0	0	0	0
128641	18	7	7	13	13	16	20	18	1	7	...	10	16	1	10	17	12	8	10	16	1
128642	10	4	18	10	10	6	16	6	3	6	...	0	0	0	0	0	0	0	0	0	0
128643	3	14	16	16	15	20	18	12	10	1	...	10	4	17	5	6	4	12	1	16	7

128644 rows × 256 columns

Figure 2. Encoded Amino Acids

	0	1	2	3
0	0.0	1.0	0.0	0.0
1	0.0	1.0	0.0	0.0
2	0.0	1.0	0.0	0.0
3	0.0	1.0	0.0	0.0
4	0.0	0.0	1.0	0.0
...	...	...	...	...
32932859	0.0	0.0	1.0	0.0
32932860	0.0	0.0	1.0	0.0
32932861	0.0	0.0	1.0	0.0
32932862	0.0	0.0	1.0	0.0
32932863	0.0	0.0	1.0	0.0

32932864 rows × 4 columns

Figure 3. One-Hot Secondary Structure Labels

C. Callbacks for Model Training

Early Stopping Callback: Stops training when the loss stops improving for a set number of epochs, preventing overfitting. With restore_best_weights=True, the model reverts to the best-performing weights.
Reduce Learning Rate (LR) on Plateau Callback: Dynamically lowers the learning rate when the loss plateaus, enabling smoother convergence during later training stages.
Model Checkpoint Callback: Saves the best model based on validation accuracy, ensuring the top-performing model is available for evaluation and deployment.

D. Custom Metric: Masked Accuracy

Since sequences contain padding, we define a custom metric (masked_accuracy) to compute accuracy only on non-padding tokens, ensuring meaningful evaluation. Masking layers are ineffective with CNNs, as they treat padding as valid input, making a custom metric necessary.

E. Model Architecture

The proposed model architecture is designed to effectively capture both local and sequential dependencies within the input data. The architecture begins with an Embedding layer, which transforms the input sequences into dense vector representations of 512 dimensions. This representation is further processed by a series of convolutional and recurrent layers to extract features at different levels of abstraction.
Convolutional Layers: Three Conv1D layers, each containing 512 filters and utilizing a kernel size of 3, are employed to capture local patterns within the sequences. These convolutional layers use the ReLU activation function, defined mathematically in (1), and are followed by MaxPooling1D layers with a pooling size of 1, which help retain feature dimensions while aggregating local information.
Recurrent Layers: To capture long-range dependencies and the sequential context of the data, the model includes both Bidirectional LSTM (Long Short-Term Memory) and Bidirectional GRU (Gated Recurrent Unit) layers, each configured with 512 units. These layers process the data in both forward and backward directions, enabling the model to consider information from both past and future time steps within the sequences. L2 regularization with a factor of 0.01 is applied to prevent overfitting by penalizing large weights.
Fully Connected Layers: After the recurrent layers, the model employs TimeDistributed Dense layers. The first two dense layers have 128 and 64 units, respectively, and use ReLU activation function.

$$ReLU(x) = max(0, x) \quad (1)$$

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (2)$$

The graph (Figure 4) shows the ReLU (Rectified Linear Unit) activation function, which outputs zero for negative inputs and the input itself for positive values. ReLU introduces non-linearity while being computationally efficient, making it widely used in deep neural networks.
Each dense layer is followed by dropout regularization with a rate of 0.6 to reduce the risk of overfitting by randomly omitting a fraction of the units during training. These layers transform the output from the recurrent layers into a lower-dimensional space. Finally, a TimeDistributed Dense layer with 4 units (corresponding to the number of classes: Padding, Coil, Helix, and Strand) and a Softmax activation as expressed in (2) is used to produce the class probabilities for each time step in the sequence. The graph (Figure 5) illustrates the Softmax function, an activation function widely used for multi-class classification tasks. It transforms a vector of raw

scores (logits) into probabilities by exponentiating each value and normalizing them so that the probabilities sum to 1. This makes the outputs interpretable as probabilities, enabling the identification of the most likely class. Softmax is particularly effective in emphasizing the dominant class while suppressing less likely ones, making it an essential component in classification models.

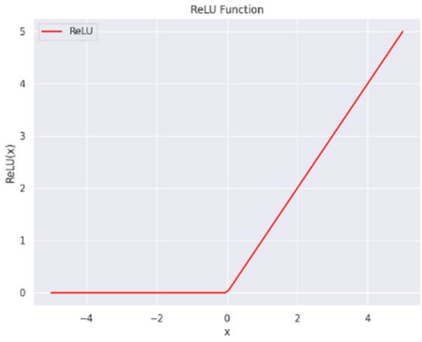

Figure 4. Rectified Linear Unit (ReLU) activation function

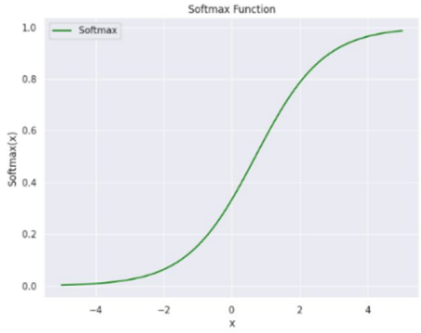

Figure 5. Softmax activation function

F. Model Compilation

The model is compiled using the categorical cross-entropy loss function (3), appropriate for multi-class classification tasks.

$$f(y, \hat{y}) = -\sum_{i=1}^{i=N} y_i \cdot log(\hat{y_i}) \quad (3)$$

Where,
$f(y, \hat{y})$ = Categorical cross-entropy loss.
y_i = True label (0 or 1) for class i from the one-hot encoded target vector.
$\hat{y_i}$ = Predicted probability for class i.
For three classes C, H and E the equation for categorical cross entropy would be:

$$f(y, \hat{y}) = -(y_C \cdot \log(\hat{y_C}) + y_H \cdot \log(\hat{y_H}) + y_E \cdot \log(\hat{y_E})) \quad (4)$$

Where,
y_C, y_H and y_E denote the true label (0 or 1) for classes C, H and E respectively.
$\hat{y_C}$, $\hat{y_H}$ and $\hat{y_E}$ denote the Predicted probability for classes C, H and E respectively.
The model was optimized with the Adam optimizer (learning rate of 0.001 which is dynamically reduced on plateau). A custom metric, masked_accuracy, is employed to ensure that padding tokens do not contribute to the accuracy calculations, thereby providing a more accurate assessment of the model's performance.

V. Results and Discussion

A. Defining Evaluation Metrics

The proposed model for protein secondary structure prediction was evaluated using a comprehensive set of metrics, including Q3 accuracy, classification report (that includes precision, recall, F1-score) and finally confusion matrix.
The individual Q3 scores of each class are given by (5), (6) and (7).

$$Q3\_C = \frac{N_{correct,\ C}}{N_C} \times 100 \quad (5)$$

$$Q3\_H = \frac{N_{correct,\ H}}{N_H} \times 100 \quad (6)$$

$$Q3\_E = \frac{N_{correct,\ E}}{N_E} \times 100 \quad (7)$$

Where,
$Q3\_C$, $Q3\_H$ and $Q3\_E$ denote the class-wise $Q3$ scores for Coil, Helix and Beta Sheet respectively.
N_C, N_H and N_E denote total number of residues in the classes Coil, Helix and Beta Sheet respectively.
$N_{correct,\ C}$, $N_{correct,\ H}$ and $N_{correct,\ E}$ denote the number of correctly predicted residues in the classes Coil, Helix and Beta Sheet respectively.

The overall Q3 score across all three classes is given by (8).

$$Q3 = \frac{(N_{correct,\ C} + N_{correct,\ H} + N_{correct,\ E})}{N_{total}} \times 100 \quad (8)$$

Where,
$Q3$ = overall Q3 score
N_{total} = Total number of residues in the dataset (across all three classes).

B. Results

The overall Q3 accuracy achieved by the model is 86.30%. The model was also tested on the CB513 dataset and achieved a Q3 accuracy of 85.29%. Comparison is done in Table IV.

TABLE IV. COMPARISON OF Q3 SCORES

	Test dataset	CB513
Q3 (overall)	0.8630	0.8529
Q3_C	0.8495	0.8462
Q3_H	0.9009	0.8921
Q3_E	0.8276	0.8068

The results from TABLE IV indicate that the model is capable of accurately predicting protein secondary structures across three main classes: C, H and E. The overall Q3 scores of 86.30% and 85.29% demonstrate that the model maintains consistent performance across different prediction tasks. Specifically, the Q3_C scores of 84.95% and 84.62% reflect a reliable ability to identify coils, while the Q3_H scores of 90.09% and 89.21% highlight the model's high accuracy in predicting helices. Additionally, the Q3_E scores of 82.76% and 80.68% show the model's capability in predicting beta-sheets, though with slightly lower accuracy compared to helices and coils. The classification report TABLE V highlights the model's strong overall performance, achieving an accuracy of 86% (86.30% to be exact) and macro/weighted averages of 0.86 for precision, recall, and F1-score. Class-wise, the model performs best on helix (H) with a precision of 0.89, recall of 0.90, and F1-score of 0.89, indicating high sensitivity and minimal false negatives. Coil (C) results show balanced precision (0.84) and recall (0.85), with a slightly higher rate of false positives. For beta strand (E), the precision is 0.87 and recall is 0.83, with an F1-score of 0.85, reflecting a slight trade-off favouring precision over recall. Overall, the model consistently predicts secondary structures well across all classes.

TABLE V. CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
C	0.84	0.85	0.84
H	0.89	0.90	0.89
E	0.87	0.83	0.85
Accuracy	-	-	0.86
Macro Avg	0.86	0.86	0.86
Weighted Avg	0.86	0.86	0.86

Confusion matrix analysis helps us to gain deeper insights into the model's performance by evaluating its ability to correctly classify residues into secondary structure classes: Coil (C), Helix (H), and Beta Strand (E). This analysis provides a detailed breakdown of true positives, false positives, and misclassifications, enabling us to identify specific areas for improvement and refine the model's predictive accuracy.

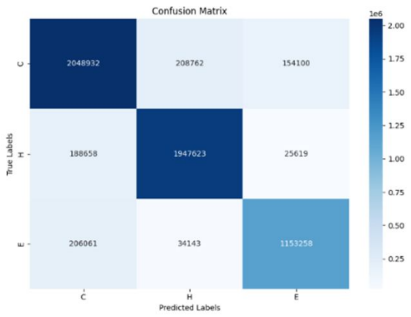

Figure 6. Confusion matrix with absolute values

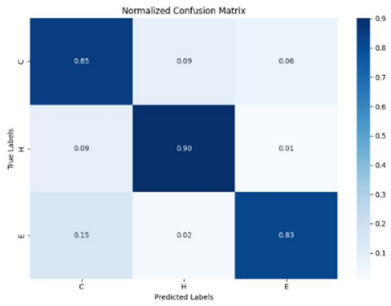

Figure 7. Confusion matrix with normalized values

The confusion matrix (Figure 6) presents the absolute number of predictions, while the normalized confusion matrix (Figure 7) provides percentage-based values for easier interpretation of the model's performance across classes. The diagonal elements represent correct predictions, and off-diagonal elements indicate misclassifications.

Coil (C): Out of 2,411,794 true coil residues, the model correctly predicted 2,048,932 (85%), while 208,762 (9%) were misclassified as helix and 154,100 (6%) as beta strand.

Helix (H): For 2,161,900 true helix residues, 1,947,623 (90%) were accurately predicted, with 188,658 (9%) misclassified as coil and 25,619 (1%) as beta strand.

Beta Strand (E): Among the 1,393,462 true beta strand residues, 1,153,258 (83%) were correctly classified, with 206,061 (15%) misclassified as coil and 34,143 (2%) as helix.

These values demonstrate that, while the model performs well overall, there are noticeable tendencies to misclassify certain residues. Misclassification is most common when distinguishing between coil and helix or between coil and beta strand. These patterns highlight specific areas where the model's performance can be improved, especially in reducing confusion between these secondary structure classes.

VI. Conclusion

The results demonstrate that the model performs consistently well across different datasets and is suitable for general protein secondary structure prediction, particularly excelling in the accurate identification of helical regions. The high Q3 accuracy scores and balanced precision, recall, and F1-scores for all three major classes—coil, helix, and beta strand—indicate that the model is reliable for practical use. Despite the strong overall performance, there is room for improvement, especially in distinguishing between coils and beta strands. The confusion matrix analysis revealed specific patterns of misclassification, suggesting that further refinements, such as improved feature engineering or exploring more sophisticated model architectures, could enhance predictive accuracy.

Future work could focus on overcoming the current limitations of the model. Notably, the sequence length was restricted to 256 residues during training and testing; lifting this restriction would allow the model to handle longer protein sequences more effectively. Additionally, expanding the model’s scope from 3-state (Q3) to 8-state (Q8) prediction, that covers even more detailed secondary structure elements, could provide deeper structural insights. Efforts to improve beta strand classification and minimize misclassification rates across all classes would further enhance the model's utility for diverse protein datasets.

References

[1] Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.Kabsch W, Sander C, Biopolymers. (1983) 22 2577-2637.

[2] Wouter G. Touw, Coos Baakman, Jon Black, Tim A. H. te Beek, E. Krieger, Robbie P. Joosten, Gert Vriend, A series of PDB-related databanks for everyday needs, Nucleic Acids Research, Volume 43, Issue D1, 28 January 2015, Pages D364–D368, https://doi.org/10.1093/nar/gku1028

[3] Reddy, M. K. (2024, November 27). amino acid. Encyclopedia Britannica. https://www.britannica.com/science/amino-acid

[4] Chou, P. Y., & Fasman, G. D. (1974). Prediction of protein conformation. Biochemistry, 13(2), 222–245. https://doi.org/10.1021/bi00699a002

[5] Garnier, J., Gibrat, J., & Robson, B. (1996). [32] GOR method for predicting protein secondary structure from amino acid sequence. Methods in Enzymology on CD-ROM/Methods in Enzymology, 540–553. https://doi.org/10.1016/s0076-6879(96)66034-0

[6] Lim VI. Algorithms for prediction of alpha-helical and beta-structural regions in globular proteins. J Mol Biol. 1974 Oct 5;88(4):873-94. doi: 10.1016/0022-2836(74)90405-7. PMID: 4427384.

[7] Dong, B., Liu, Z., Xu, D., Hou, C., Dong, G., Zhang, T., & Wang, G. (2024). SERT-StructNet: Protein secondary structure prediction method based on multi-factor hybrid deep model. Computational and structural biotechnology journal, 23, 1364–1375. https://doi.org/10.1016/j.csbj.2024.03.018

[8] V. Bongirwar and A. S. Mokhade, "A Hybrid Bidirectional Long Short-Term Memory and Bidirectional Gated Recurrent Unit Architecture for Protein Secondary Structure Prediction," in IEEE Access, vol. 12, pp. 115346-115355, 2024, doi: 10.1109/ACCESS.2024.3444468.

[9] Yang, S., & Chen, X. (2024). Prediction of Protein Secondary Structure Using a Hybrid Convolutional Blocks with GRU Units. Frontiers in Computing and Intelligent Systems, 10(3), 23-30. https://doi.org/10.54097/k8rmxp27

[10] M. M. Mohamed Mufassirin, M. A. H. Newton, J. Rahman and A. Sattar, "Multi-S3P: Protein Secondary Structure Prediction With Specialized Multi-Network and Self-Attention-Based Deep Learning Model," in IEEE Access, vol. 11, pp. 57083-57096, 2023, doi: 10.1109/ACCESS.2023.3282702.

[11] Ema, R. R., Khatun, M. A., Adnan, M. N., Kabir, S. S., Galib, S. M., & Hossain, M. A. (2022). Protein Secondary Structure Prediction based on CNN and Machine Learning Algorithms. International Journal of Advanced Computer Science and Applications, 13(11). https://doi.org/10.14569/ijacsa.2022.0131108
[12] Jin, X., Guo, L., Jiang, Q., Wu, N., & Yao, S. (2022). Prediction of protein secondary structure based on an improved channel attention and multiscale convolution module. Frontiers in Bioengineering and Biotechnology, 10. https://doi.org/10.3389/fbioe.2022.901018
[13] Lyu, Z., Wang, Z., Luo, F., Shuai, J., & Huang, Y. (2021). Protein secondary structure prediction with a reductive deep learning method. Frontiers in Bioengineering and Biotechnology, 9. https://doi.org/10.3389/fbioe.2021.687426
[14] Sutanto, V., Sukma, Z., & Afiahayati, A. (2020). Predicting secondary structure of protein using hybrid of convolutional neural network and support vector machine. International Journal of Intelligent Engineering and Systems, 14(1), 232–243. https://doi.org/10.22266/ijies2021.0228.23
[15] Ma, Y., Liu, Y. & Cheng, J. Protein Secondary Structure Prediction Based on Data Partition and Semi-Random Subspace Method. Sci Rep 8, 9856 (2018). https://doi.org/10.1038/s41598-018-28084-8