Transformer Encoder with Protein Language Model for Protein Secondary Structure Prediction

Ammar Kazm
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia | College of Education for Pure Sciences, Wasit University, Iraq
awadkazm@graduate.utm.my (corresponding author)

Aida Ali
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia
aida@utm.my

Haslina Hashim
School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia
haslinah@utm.my

Received: 2 January 2024 | Revised: 15 January 2024 | Accepted: 17 January 2024

 DOI: https://doi.org/10.48084/etasr.6855

ABSTRACT

In bioinformatics, protein secondary structure prediction plays a significant role in understanding protein function and interactions. This study presents the TE_SS approach, which uses a transformer encoder-based model and the Ankh protein language model to predict protein secondary structures. The research focuses on the prediction of nine classes of structures, according to the Dictionary of Secondary Structure of Proteins (DSSP) version 4. The model's performance was rigorously evaluated using various datasets. Additionally, this study compares the model with the state-of-the-art methods in the prediction of eight structure classes. The findings reveal that TE_SS excels in nine- and three-class structure predictions while also showing remarkable proficiency in the eight-class category. This is underscored by its performance in Qs and SOV evaluation metrics, demonstrating its capability to discern complex protein sequence patterns. This advancement provides a significant tool for protein structure analysis, thereby enriching the field of bioinformatics.

Keywords-protein secondary structure prediction; bioinformatics; nine-class protein prediction; transformer model; Ankh protein language model

I. INTRODUCTION

Proteins are made up of chains of amino acids. By altering their arrangement, 20 different types of acids can create a wide range of proteins. The primary structure of a protein is represented by one sequence, comprising the specific order in which the amino acids are arranged [1], and is referred to as 1D structure. Tertiary structures, often referred to as three shapes, are formed in living organisms through the interactions, among amino acids. These interactions play a crucial role in determining the function of proteins [2]. To fully comprehend the relationship between the tertiary structures of a protein, it is important to predict its secondary structure [3]. The use of efficient techniques to forecast protein structures has become essential in closing the disparity between the number of recognized protein sequences and the determined structures due to the limitations that experimental procedures entail, such as time requirements and the substantial costs involved [4]. These predictive models are instrumental in enhancing our comprehension of protein functions and may be utilized in applications like drug development and disease control [5].

Secondary structure in proteins refers to the folded patterns that occur within a chain of acids as a result of forces like hydrogen and van der Waals bonds. To precisely define structure, the Dictionary of Secondary Structure of Proteins (DSSP) was devised [6]. This program analyzes the coordinates of proteins with known structures to identify patterns of hydrogen bonding and geometric characteristics. DSSP assigns a type of secondary structure to each residue in the protein. The original classification consisted of eight classes: G (310 helix), H (α-helix), I (π-helix), B (isolated β-strand), E (extended strand), S (bend), T (turn), and L (irregular structure). These categories are often grouped into three group classes: H, G, I to

helix (H), (B, E) to strand (E), and T, S, L to coil (C). The latest iteration of DSSP, version 4.0, which was released in 2021, marks a significant update in the field of protein secondary structure classification. This version extends the conventional eight types of secondary structures to include a ninth type, known as the poly-proline helix (P) [7]. The task of Protein Secondary Structure Prediction (PSSP) involves assigning classes of structures, such as alpha helices, beta sheets, and coils to each individual amino acid in a protein chain. For computational methods to predict a structure, it is necessary to represent acids as numeric vectors. One-hot vector approach uses 21-encoding for each amino acid in protein sequence, which includes the 20 standard amino acids that make up the proteins and 1 non-standard amino acid represented by X to indicate an unknown or unspecified amino acid. However, this method has shown limited accuracy, in prediction. Another used technique involves utilizing PSSM profile features [8] or HHM profile features [9]. These profile features incorporate information derived from analyzing sequence alignments obtained from a large protein sequence database.

Creating Hidden Markov Models (HMMs) or Position Specific Scoring Matrices (PSSMs) for each template sequence can be a time-consuming process, especially when dealing with proteins that have no sequences. To overcome this hurdle, recent advancements have introduced novel protein representation techniques inspired by methods used in natural language processing [10-13]. These techniques involve the usage of pretrained protein language models, followed by fine tuning for specific tasks. These models can achieve performance even with limited task specific data available, where embedding from a language model pretrained on a large corpus of protein sequences effectively replaces evolutionary information. The implementation of this approach has demonstrated encouraging outcomes in several protein-related subsequent studies [4, 7, 14-19]. In the early stages of PSSP research statistical approaches were predominantly used. These methods focused on determining the likelihood of amino acids, in protein structures [20]. Initially these predictors were designed for a three-class secondary structure prediction due to training data availability and computational constraints of that time. However, the particular methods encountered challenges in achieving high Q3 accuracy because they struggled to extract information from primary protein structure sequences. To overcome this limitation and improve their performance, researchers started incorporating information and position specific scoring matrices into the prediction process. This advancement proved significant, leading to a Q3 accuracy exceeding 70% [21].

Various machine learning techniques have been used for performing coarse-grained prediction, including decision trees [22], support vector machines [23], Neural Networks (NN) [24], HMMs [25], probabilistic graph models [26], and k-nearest neighbors [27]. The methodologies in this area primarily utilize a fixed-size sliding window approach. This method was employed to forecast the secondary structure category of the essential amino acid residue in a given sequence. JPred4 [28] and PSIPRED V3.0 [29] were notable among the initial prediction algorithms. These techniques laid the foundation for further progress in the field, demonstrating the effectiveness of machine learning in understanding and predicting protein structures. The increased availability of data has led to the dominance of sequence-to-sequence deep model predictions, which have achieved state-of-the-art performance. Innovations in this area include DCRNN [30], which uses cascaded Convolutional and Recursive NN to extract both multiscale local and global contextual features. Other significant contributions include multiscale chained convolutional architecture for improved eight-state prediction [31]. SPIDER3 [32] uses LSTM BRNNs to capture complex amino acid interactions, DeepACLSTM [33] integrates networks with LSTM units and utilizes specific dimensions in protein sequence feature vectors. MUFOLD SS [34] and SAINT [35] both employ Deep inception-inside-inception networks with MUFOLD SS emphasizing inception modules while SAINT incorporates self-attention mechanisms. SPOD 1D [36] combines LSTM BRNN and ResNet models with residue contact maps for its predictions. NetSurfP 2.0 [37] employs convolutional and LSTM networks, while ShuffleNet_SS [5] focuses on a lightweight convolutional NN. Another important development is the introduction of the protein encoder [38]. This method employs a two-step process, beginning with an unsupervised autoencoder for feature extraction, followed by an ensemble of feature selection methods. A common element in earlier prediction models is their reliance on profile features, which are primarily obtained from Multiple Sequence Alignments (MSA). Nevertheless, the specific process, especially considering the rapidly expanding protein sequence databases, poses a significant time constraint. In response to this challenge, recent research has shifted toward leveraging embedding features extracted from pretrained protein language models. For instance, DML_SS [4] applies learning through a deep centroid model, for its predictions. SPOT 1D LM [19] synergizes embeddings from language models with one hot encoding techniques. LIFT SS [7] focuses on tuning pretrained protein language models.

In the field of predicting protein secondary structures, it is interesting to note that most current predictors apart from LIFT_SS [7] rely on the eight class assignments of structures from the previous version of the DSSP program for their training and evaluation data. Notably, even subsequent studies published after the introduction of DSSP 4.0 have continued to rely on eight-class secondary structure information, rather than adopting the more recent nine-class secondary structure classification. This trend indicates that these methodologies are being trained and evaluated using potentially outdated labeling information. In this study, the latest edition DSSP 4, a comprehensive database for secondary structure sequences, was utilized. The use of DSSP 4 ensured the training and evaluation data were based on a detailed classification of protein structures. Additionally, the Ankh protein language model [39] was adopted for obtaining protein embeddings, leveraging its capability to accurately represent protein sequences and replace the need for more computationally intensive evolutionary information. TE_SS, a deep transformer-based model [40], specifically designed to discern complex relationships between distant and proximal amino acid sequences in proteins, is proposed. This model is specifically

designed to discern complex relationships between local and nonlocal amino acid sequences in proteins, processing sequential features in parallel, in contrast to existing models that extract features sequentially. The architecture of this model enables it to capture patterns and interactions within protein structures, enhancing the prediction of secondary structure.

II. METHODS

A. Dataset

In this research, a collection of protein training data was generated using the PISCES server [38]. The latter is well known for creating curated lists of sequence subsets from the Protein Data Bank (PDB). To assess protein structure prediction algorithms, criteria and parameters related to sequence identity were applied. The PISCES server utilizes a filter based on predefined protein parameters. Subsequently it sends the resulting lists and sequence files directly to the email address provided. To ensure the dataset's reliability and usefulness, the PISCES server was configured with settings, including a maximum resolution of 2.0 Å, an upper limit R value of 2.0 and a requirement that there will be no more than 50% sequence identity between any pair of protein sequences. Initially the dataset suggested consisted of 16,225 proteins. However, 188 proteins from this collection were excluded because they lacked corresponding information. In addition, to maintain the integrity of the performed analysis and avoid data contamination, any proteins that overlapped with the proposed test dataset were eliminated. Following these rigorous filtering criteria, a refined dataset, labeled as 16,037, consisting of 16,037 proteins was successfully curated. This dataset was strategically partitioned, with 15,037 proteins designated for the training set and the remaining 1000 proteins allocated for validation purposes.

This study involves datasets for secondary structure analysis of protein based on the 9-class classification provided by DSSP4. The DSSP software generates a DSSP file for each protein with an established structure, which contains detailed secondary structure information derived from the protein's three-dimensional structural data recorded in the PDB database. In the performed methodology, the Biopython library was initially employed to retrieve the PDB file corresponding to a given protein chain. This file is accessed from the PDB website using the specific PDB ID and the chain ID of the protein chain. The occurrence of nonstandard amino acids in these files, including modified residues was observed. A notable example includes the representation of methionine (MET) and selenomethionine (MSE) by the one-letter code M. To address this issue, a conversion process in which the three-letter amino acid codes in the PDB file were translated to their one-letter equivalents, with nonstandard amino acids denoted as X, was implemented. This modified sequence is referred to as the target primary sequence. Subsequently, the identical DSSP file was acquired on the basis of the PDB file. Contiguous fragments of amino acid residues and their associated secondary structures were extracted from this file, guided by the chain ID and the residue sequence number. However, extracting a primary sequence from the DSSP file that exactly matches the target primary sequence in terms of sequence composition or length is often not feasible [6]. To accurately represent the sequence of protein structure, the primary sequence of interest is matched with the sequence obtained from the DSSP file. During this alignment process, any gaps that occur are filled with the letter X to indicate unassigned types of structures. To achieve this alignment, the Pairwise2 alignment algorithm from the BioPython package was utilized [39].

Performance metrics for nine-class prediction were assessed on diverse datasets. Three editions of the CASP competition, namely CASP12, CASP13, and CASP14, were utilized. These datasets encompass a selection of 47, 41, and 33 protein chains, respectively, carefully chosen to represent real-world challenges in protein structure prediction. Additionally, the CB433 test data [4], a curated and filtered subset of the widely used CB513 dataset, comprising 433 protein structures was considered. Evaluating the proposed model fairly against existing models necessitates the use of datasets that adhere to the same 8-class system. For this purpose, two well-known test datasets, TEST2016 and TEST2018 [34], containing 1213 and 250 protein sequences accordingly, were chosen. These are in line with the training and validation sets, which include 10029 and 983 proteins, respectively. Across all four datasets, the maximum length of any protein sequence does not exceed 700. Additionally, the primary and secondary structure sequence data, which is the standard for these datasets, is also utilized.

B. Embedding

Pretrained models that focus on protein language (Protein Language Models-PLMs) have become a tool, in biological applications serving as a strong foundation for modeling protein related tasks. While most approaches rely on these models for extracting features, this study takes an approach by utilizing the Ankh model [39], which is a large unsupervised PLM. Ankh, built on a transformer-based architecture and has been trained on the BFD [41] and UniRef50 [42] dataset. It achieves state of the art performance while using less than 10% of the parameters compared to models. This impressive efficiency opens up possibilities for accessible and scalable protein modeling applications. One of the strengths of the Ankh model lies in its ability to extract high quality embedding features that represent proteins accurately. These features are representations of protein sequences that capture information about their structure, function and evolutionary relationships. To acquire the embedding feature of a specific protein chain using Ankh we input its sequence into the model encoder and retrieve its output. Each amino acid, in a protein sequence is assigned a 1536-feature vector through the output embedding, which captures its information. For every protein sequence L this model generates an embedding vector of size (L*1536). The embeddings acquired from the embedding process were used as the input for the model.

C. Model Architecture

A novel model has been developed to predict protein structures. It heavily relies on transformer architectures. These transformers are great, at identifying both distant relationships within protein sequences by utilizing self-attention mechanisms and feed forward layers. The initial input for this model is a two-dimensional embedding, with (L, 1536) sequence

dimensions. This embedding is generated using the training Ankh model. Additionally, the model incorporates encoding for each amino acid in the sequence to enhance information representation. After that, a series of N transformer encoders process the input as shown in Figure 1. This approach demonstrates how effective the model is at capturing patterns, within protein sequences leading to accurate predictions of protein structures.

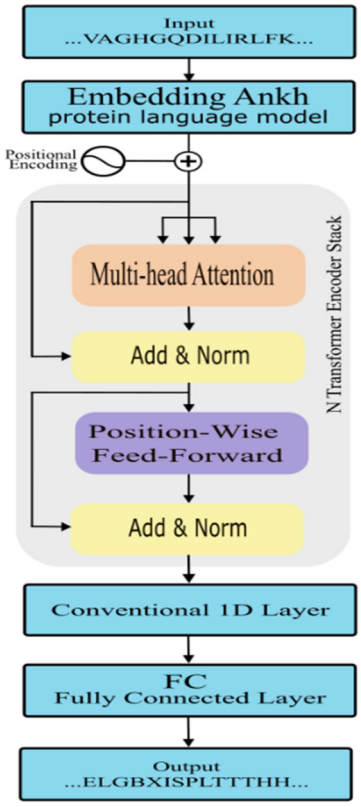

Fig. 1. Transformer-based model architecture for protein secondary structure prediction.

1) Positional Encoding

To ensure the proposed model effectively takes into account the nature of protein sequences, positional encoding was incorporated. This approach produces data on the precise locations of amino acids throughout the protein sequence. By combining positional encoding with amino acid embeddings, not only can this model comprehend the unique characteristics of each amino acid, but also their contextual relationships within the sequence. This approach is crucial for capturing the spatial details of amino acids, which are essential for accurately predicting protein secondary structure. Positional encoding (PESS) is defined as follows:

$$\text{PESS}_{(\text{psn},2i)} = \sin\left(\text{psn}/10000^{2i/dim_{model}}\right) \quad (1)$$

$$\text{PESS}_{(\text{psn},2i+1)} = \cos\left(\text{psn}/10000^{2i/dim_{model}}\right) \quad (2)$$

where psn represents the position of an amino acid in the sequence and i is its dimension in the encoding space, whereas dim_{model} refers to the dimensionality of the model [43]. In (1) the encoding for odd sequence positions is addressed, while (2) pertains to the encoding for even sequence positions.

In (3) the positional encoding obtained from (1) and (2) is illustrated and added into the input embeddings.

$$Xemb_i' = Xemp_i + \text{PESS} \quad (3)$$

where $Xemb_i'$ is positionally encoded embedding for i amino acid and $Xemp_i$ is i amino acid's embedding obtained from the Ankh model.

By including these data the suggested model acquires a comprehension of the protein's arrangement, which improves its predictive abilities, for the secondary structure.

2) Transformer Encoder in Protein Secondary Structure Prediction

The transformer encoder is a component of the transformer architecture [43] used for processing sequences in parallel. It is composed of layers, each of which has two sublayers: the Position-Wise Feed Forward Network and the Multi Head Self Attention Mechanism.

The key elements of the Transformer Encoder are:

a) Multi-head Attention

The multi head self-attention mechanism plays a role, in the encoder by allowing the model to evaluate and adjust the importance of segments within an input sequence. It creates three vector representations, i.e. query (Q), key (K), and value (V) for each input element. By measuring the similarity between Q and K, the attention scores are calculated to determine a sum of V vectors highlighting most relevant information. This process is performed across multiple heads enabling focus on different aspects of the sequence. The mathematical formulation, for this process is [43]:

$$Attention\ (Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (4)$$

where $\sqrt{d_K}$ serves as a scaling factor to ensure values for sequences.

b) Feed–Forward Networks

After the self-attention mechanism, the data pass through a feed-forward NN, which is applied to each position separately and identically. This network consists of fully connected layers with activation functions and is responsible for further transforming the representation.

c) Layer Normalization

Each sublayer of both self-attention and feed-forward networks in the transformer encoder has a residual connection around it, followed by layer normalization. The residual connections help mitigate the vanishing gradient problem, enabling the training of very deep models.

d) Stacking of Layers

The self-attention, multi-head attention, and feed-forward layers are stacked together, forming multiple encoder layers. Each layer builds upon the previous one, gradually extracting increasingly complex and higher-level representations of the sequence.

3) Convolution 1D Layer

To further augment the model's capability to extract informative features, a 1D convolutional layer follows the transformer encoder. This layer operates along the feature dimension, applying learnable filters to capture local patterns and dependencies within the feature space. Mathematically, the expression of the one-dimensional convolution operation can be formally articulated as:

$$Y = f(w \circledast X + b) \quad (5)$$

where Y is the output feature map, X is the input feature map, w is the convolutional filter, b is the bias term, $\circledast$ is the convolution operator, and f is the activation function.

4) Final Fully Connected Layer

The architecture concludes with a fully connected layer, which serves as the classification component of the model. This layer translates the processed features into predictions of the protein's secondary structure.

D. Evaluation Metrics

To evaluate the effectiveness of the proposed approach, two employed metrics were utilized; Q_s accuracy and Segment Overlap (SOV) [44]. Q_s accuracy measures how the predicted secondary structure aligns, with the determined secondary structure specifically looking at the proportion of residues that match. Meanwhile, SOV assesses how closely the predicted and experimentally determined secondary structure segments resemble each other. In addition to these metrics, F1, Precision, and Recall were also employed to evaluate the proposed model's performance on the selected test dataset. Q_s accuracy quantifies the proportion of residues where the predicted secondary structure aligns with the findings. This metric plays a role in assessing a model's ability to accurately classify types of secondary structures found in proteins. Precision is indicated by how residues are correctly predicted for their corresponding secondary structures. It expands on the conventional Q_3 accuracy measure S = (H, E, C) by categorizing secondary structures into nine categories: S = (H, G, I, P, B, E, T, S, L). To compute Q_s we divide the number of correctly predicted residues, in state s (n_s) by the total number of residues actually in state s (N_s), with s representing each state within the set S. This is formally represented in (6):

$$Q_s = \frac{n_s}{N_s} \times 100 \ , s \in S \quad (6)$$

To calculate the overall accuracy for per residue prediction all (n_s) values for each state s in set S are summed up and divided by the sum of all (N_s) values for each state s in set S [4]:

$$Q_{|S|} = \frac{\sum_{s \in S} n_s}{\sum_{s \in S} N_s} \times 100 \quad (7)$$

The SOV metric is crucial when evaluating the precision of protein secondary structure predictions. Unlike accuracy measures, SOV provides a detailed evaluation by considering both length and overlap between the predicted and actual segments. This metric is useful when assessing predictions for structure elements like alpha helices and beta sheets which can vary significantly in length. SOV compares how well predicted segments align with segments in terms of length and overlap. It takes into account variations, in segment size making it a comprehensive and realistic measure to assess prediction performance for complex proteins that exhibit diverse secondary structures.

E. Implementation Details

PyTorch framework was used as it offers a graph, imperative execution style and a wide range of tools and libraries. To ensure training and avoid overfitting to data patterns, the minibatch size was set to 8 and random sampling was employed to create minibatches. For optimizing the suggested models, the AdamW optimizer was used with a weight decay value of 0.0001. Throughout the training process a fixed learning rate of 0.00005 was maintained. To enhance the proposed model's performance, a custom cross loss function that handles class imbalances by allowing optional weights, for different classes was implemented. This function calculates the loss for each instance without reduction, and then averages it across the minibatch while considering the provided class weights. This approach ensures an impact of each class on the models learning process. Moreover, a stopping criterion was implemented. The particular criterion halts training if there is no improvement in Q_s accuracy, on the validation set for 5 consecutive epochs. This study experiments were conducted using an NVIDIA Tesla V100 GPU with 16 GB VRAM and 32 GB system memory. The transformer encoder architecture used in this study consisted of 5 layers, each equipped with 8 attention heads. In these layers, a dropout rate of 0.2 was incorporated. The dimension of the feed forward network was set to 2048. The model's convolutional layers produced an output with 1024 channels.

III. RESULTS

A. Ablation Study

We comprehensively evaluated the performance of the proposed method through a series of experiments on the CB433 test set and our validation set. These experiments were meticulously designed to analyze the influence of key hyperparameters, specifically the number of transformer encoder layers, the number of attention heads, and the learning rate, on the model's effectiveness.

1) Number of Encoder Layers

To examine the effect of encoder layer depth on model performance, this study experimented with architectures ranging from 1 to 7 layers, each coupled with a fixed configuration of 8 attention heads. The validation and test results on the CB433 dataset, as depicted in Figure 2, indicate that the architecture with 5 encoder layers achieved the highest 9-class accuracy (Q9). Deeper models can learn more complex contextual representations and better capture long-range dependencies in protein sequences, performance plateaus, but this ability slightly declines beyond 5 layers. This could be indicative of overfitting or vanishing gradients, affecting the model's generalizability and learning efficacy.

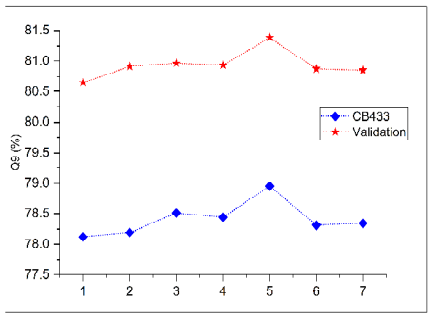

Fig. 2. Performance evaluation of transformer models with varying numbers of encoder layers.

2) Number of Attention Heads

To investigate the optimal configuration for protein secondary structure prediction, there was a focus on the number of attention heads in the transformer encoder layer. Configurations with 1, 2, 3, 4, 6, 8, 12, and 16 attention heads were tested for their impact on the model's performance to be determined. As shown in Figure 3, the model with 8 attention heads proved most effective in both validation and testing sets, particularly on the CB433 dataset, achieving significant improvements in 9-class accuracy, pointing to the optimal balance between the granularity and breadth of attention mechanisms. While increasing the number of attention heads generally improves model's ability to discern intricate relationships within protein structures, a threshold exists beyond which additional heads may not enhance or could even reduce predictive accuracy. This highlights the importance of fine-tuning attention mechanisms in transformer models for specialized bioinformatics tasks mechanisms in transformer models for specialized bioinformatics tasks.

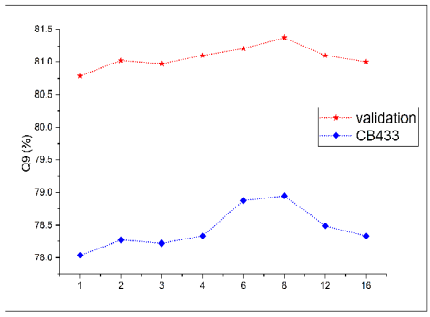

Fig. 3. Comparative analysis of prediction accuracy across different numbers of attention heads.

3) Learning Rate

The study examined the best hyperparameters for protein secondary structure prediction and found that the learning rate had a significant impact on model accuracy. The former rigorously assessed the model's performance throughout a range of learning rates, as shown in Figure 4: 0.001, 0.0005, 0.0001, 0.00005, 0.00001, and 0.000005. In the testing and validation stages, a learning rate of 0.00005 produced the best 9-class accuracy, especially when employing the CB433 dataset. Interestingly, there was a clear trend in the model's performance: the accuracy decreased dramatically at higher learning rates (0.001 and 0.0005) pointing the detrimental effect of rapid weight adjustments. However, as the learning rate was gradually reduced, a notable improvement in accuracy was observed, culminating in the optimal performance at 0.00005.

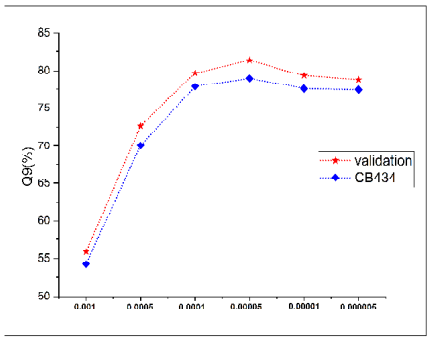

Fig. 4. Impact of learning rate on model accuracy.

B. Comparative Analysis on Eight-State Prediction

This section provides a comparative analysis of the proposed method against a selection of state-of-the-art predictors methods, specifically focusing on two distinct test data called, TEST2016 and TEST2018. To ensure an equitable comparison, data for existing predictors were sourced from the literature [4, 7, 9]. The comparison encompasses a variety of methods: 10 predictors based on profile features and 3 predictors based on embedding features (Table I). For the LIFT_SS method, the most accurate results were selected from three lightweight fine-tuning approaches. In the conducted analysis the TE_SS method was evaluated, against these 13 predictors through the employment of different metrics on two test datasets. These metrics include Q8 and SOV8 for predictions in 8 classes and Q3 and SOV3 for predictions in 3 classes. The detailed results can be found in Table I, which showcases the performance of the proposed method alongside the 13 methods for each metric. Table I clearly demonstrates that the TE_SS method outperforms the others in predicting protein secondary structures in both 8- and 3- class formats. Not only does this comprehensive analysis reveal the strength of the TE_SS model, but also its advancement over existing state of the art methods setting a new standard, in protein secondary structure prediction.

C. Comparative Analysis on Nine-State Prediction

To assess the effectiveness of the TE_SS framework experiments were conducted using four widely used benchmark datasets, in the field of protein structure analysis. TE_SS was compared against two leading methods for predicting 9 class protein structures; DML_SSembed and LIFT_SS. Both these methods utilize embeddings derived from ProtTrans, a trained PL). These two methods were selected based on their utilization of the 9 class predictor from DSSP4.

DML_SSembed employs a centroid model for sequence to sequence prediction. It assigns a centroid in the embedding

space to each structure category and aims to maximize the similarity between each amino acid and its corresponding centroid. This approach enhances the accuracy of secondary structure prediction. In contrast, LIFT_SS utilizes a fine tuning strategy on the pre trained PLM by employing 7 state of the art fine tuning techniques. This enables LIFT_SS to predict structures accurately by introducing new parameters during the embedding process. The results of these comparisons including predictions, for both 9- and 3- class scenarios are presented in Tables II and III. Notably, the highest metric values were taken from the 7 fine-tuning techniques used by LIFT_SS. The data for existing predictors were obtained from [7]. It is worth mentioning that the TE_SS model consistently outperformed both DML_SSembed and LIFT_SS exhibiting its accuracy and effectiveness, in predicting protein structure.

TABLE I. COMPARISON (Q8, Q3, SOV8, AND SOV3 ACCURACY) WITH STATE-OF-THE-ART METHODS

Method	TEST2016				TEST2018			
	Q8	SOV8	Q3	SOV3	Q8	SOV8	Q3	SOV3
CNN_BIGRU [45]	73.91	70.92	85.04	81.61	72.78	68.75	84.17	79.41
DeepACLSTM [33]	75.19	73.67	85.62	82.6	73.42	71.32	84.66	80.05
DCRNN [30]	72.19	68.63	83.72	78.39	70.6	65.82	82.75	75.1
DeepCNN [31]	74.54	71.56	85.14	79.31	72.75	69.18	84.16	76.83
MUFold-SS [34]	76.03	73.67	85.97	81.98	74.29	71	84.63	79.53
NetSurfP-2.0 [37]	-	-	-	-	73.81	71.14	85.31	78.58
SPOD-1D [36]	76.03	73.88	86.67	79.52	74.26	71.45	85.66	78.77
SAINT [35]	76.23	-	-	-	74.48	-	-	-
SPIDER-3 [32]	-	-	84.66	75.62	-	-	83.84	73.89
DML_SS [4]	76.62	74.6	86.1	82.72	74.82	72.23	84.83	80.5
SPOT-1D-LM [19]	-	-	-	-	76.47	-	86.74	-
DML_SSembed [4]	78.03	75.9	87.41	84.51	76.48	73.44	86.82	82.43
LIFT_SS [7]	78.7	76.79	87.84	84.76	76.86	74.24	87.13	82.32
TE_SS	**79.08**	**77.02**	**87.99**	**84.81**	**77.57**	**74.60**	**87.31**	**82.47**

TABLE II. COMPARATIVE 9-CLASS PSSP RESULTS ON THE TEST DATASETS

Method	CASP12		CASP13		CASP14		CB433	
	Q9	SOV9	Q9	SOV9	Q9	SOV9	Q9	SOV9
DML_SSembed [4]	74.81	70.07	72.32	67.25	65.4	58.43	75.59	73.35
LIFT_SS [7]	75.82	70.81	73.07	68.06	66.59	59.71	76.87	74.36
TE_SS	**76.44**	**71.03**	**75.62**	**71.51**	**67.47**	**60.29**	**78.95**	**76.37**

TABLE III. COMPARATIVE 3-CLASS PSSP RESULTS ON THE TEST DATASETS

Methods	CASP12		CASP13		CASP14		CB433	
	Q3	SOV3	Q3	SOV3	Q3	SOV3	Q3	SOV3
DML_SSembed [4]	84.56	78.48	82.75	74.99	77.7	68.82	85.9	80.63
LIFT_SS [7]	85.24	79.38	83.56	77.3	78.38	68.23	86.69	81.28
TE_SS	**85.55**	**79.84**	**84.69**	**77.50**	**78.56**	**69.71**	**87.35**	**81.64**

D. Multi-Metric Evaluation

To thoroughly evaluate models performance, an approach was adopted by considering evaluation metrics, such as F1 score, Precision, and Recall. These metrics were applied to CB433, CASP12, CASP13, and CASP14 datasets. This rigorous evaluation strategy ensured that the model's effectiveness in predicting protein structure was reliable and applicable to a range of protein sequences. Table IV provides a summary of the models performance on these metrics highlighting its accuracy in predicting protein structure. The proposed model consistently performed satisfactorily across all datasets indicating its potential, for various protein structure prediction tasks.

TABLE IV. PERFORMANCE OF TE_SS ON TEST DATASETS

Dataset	F1	Precision	Recall
CASP12	61.28	68.66	57.45
CASP13	58.95	64.24	56.56
CASP14	45.11	53.06	42.94
CB433	67.72	73.16	65.55

IV. CONCLUSIONS

In this study, the effectiveness of the transformer-based TE_SS model in predicting protein structures has been demonstrated. Utilizing the Ankh protein language model for feature embedding, the TE_SS model achieves accurate predictions of protein structures in both nine and eight classification systems. The model's performance in predicting 9-class structures was evaluated on CASP12, CASP13, CASP14, and CB433 test datasets. Also, the model, trained on data containing 8 classes, was evaluated on two publicly available test datasets, TEST2016 and TEST2018. The experimental results indicate improved accuracy compared to the other models. A notable advancement of TE_SS is its adeptness in capturing both short-range and long-range dependencies among residues in proteins. The ability of this transformer-based model to process sequence data in parallel demonstrates its efficiency and effectiveness in analyzing complex protein structures. However, it is worth noting that the proposed method has limitations in terms of its demanding resources and GPU memory requirements. Moreover, the

model currently lacks the ability to provide information regarding the reliability or confidence level of its predictions. This shortcoming is especially evident when the model encounters specific types of proteins or disordered regions within proteins, where its predictions may be less accurate or reliable. For future work, it is imperative to address these limitations, potentially by developing methods to estimate prediction reliability and optimizing the model for reduced resource consumption.

REFERENCES

[1] S. Damodaran and K. L. Parkin, Eds., "Amino Acids, Peptides, and Proteins," in *Fennema's Food Chemistry*, 5th ed., Boca Raton, FL, USA: CRC Press, 2017.

[2] S. Tahzeeb and S. Hasan, "A Neural Network-Based Multi-Label Classifier for Protein Function Prediction," *Engineering, Technology & Applied Science Research*, vol. 12, no. 1, pp. 7974–7981, Feb. 2022, https://doi.org/10.48084/etasr.4597.

[3] M. Zubair *et al.*, "A Deep Learning Approach for Prediction of Protein Secondary Structure," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 3705–3718, Mar. 2022, https://doi.org/10.32604/cmc.2022.026408.

[4] W. Yang, Y. Liu, and C. Xiao, "Deep metric learning for accurate protein secondary structure prediction," *Knowledge-Based Systems*, vol. 242, Apr. 2022, Art. no. 108356, https://doi.org/10.1016/j.knosys.2022.108356.

[5] W. Yang, Z. Hu, L. Zhou, and Y. Jin, "Protein secondary structure prediction using a lightweight convolutional network and label distribution aware margin loss," *Knowledge-Based Systems*, vol. 237, Feb. 2022, Art. no. 107771, https://doi.org/10.1016/j.knosys.2021.107771.

[6] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983, https://doi.org/10.1002/bip.360221211.

[7] W. Yang, C. Liu, and Z. Li, "Lightweight Fine-tuning a Pretrained Protein Language Model for Protein Secondary Structure Prediction." bioRxiv, Mar. 23, 2023, https://doi.org/10.1101/2023.03.22.530066.

[8] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices11Edited by G. Von Heijne," *Journal of Molecular Biology*, vol. 292, no. 2, pp. 195–202, Sep. 1999, https://doi.org/10.1006/jmbi.1999.3091.

[9] S. R. Eddy, "Profile hidden Markov models.," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, Jan. 1998, https://doi.org/10.1093/bioinformatics/14.9.755.

[10] A. Rives *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, Apr. 2021, Art. no. e2016239118, https://doi.org/10.1073/pnas.2016239118.

[11] A. Elnaggar *et al.*, "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7112–7127, Jul. 2022, https://doi.org/10.1109/TPAMI.2021.3095381.

[12] Z. Lin *et al.*, "Language models of protein sequences at the scale of evolution enable accurate structure prediction." bioRxiv, Jul. 21, 2022, https://doi.org/10.1101/2022.07.20.500902.

[13] B. Ahmed, G. Ali, A. Hussain, A. Baseer, and J. Ahmed, "Analysis of Text Feature Extractors using Deep Learning on Fake News," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7001–7005, Apr. 2021, https://doi.org/10.48084/etasr.4069.

[14] J. Singh, T. Litfin, J. Singh, K. Paliwal, and Y. Zhou, "SPOT-Contact-LM: improving single-sequence-based prediction of protein contact map using a transformer language model," *Bioinformatics*, vol. 38, no. 7, pp. 1888–1894, Mar. 2022, https://doi.org/10.1093/bioinformatics/btac053.

[15] H. Stark, C. Dallago, M. Heinzinger, and B. Rost, "Light attention predicts protein location from the language of life," *Bioinformatics Advances*, vol. 1, no. 1, Jan. 2021, Art. no. vbab035, https://doi.org/10.1093/bioadv/vbab035.

[16] S. Pokharel, P. Pratyush, M. Heinzinger, R. H. Newman, and D. B. Kc, "Improving protein succinylation sites prediction using embeddings from protein language model," *Scientific Reports*, vol. 12, no. 1, Oct. 2022, Art. no. 16933, https://doi.org/10.1038/s41598-022-21366-2.

[17] A. Villegas-Morcillo, A. M. Gomez, and V. Sanchez, "An analysis of protein language model embeddings for fold prediction," *Briefings in Bioinformatics*, vol. 23, no. 3, May 2022, Art. no. bbac142, https://doi.org/10.1093/bib/bbac142.

[18] M. H. Hoie *et al.*, "NetSurfP-3.0: accurate and fast prediction of protein structural features by protein language models and deep learning," *Nucleic Acids Research*, vol. 50, no. W1, pp. W510–W515, Jul. 2022, https://doi.org/10.1093/nar/gkac439.

[19] J. Singh, K. Paliwal, T. Litfin, J. Singh, and Y. Zhou, "Reaching alignment-profile-based accuracy in predicting protein secondary and tertiary structural properties without alignment," *Scientific Reports*, vol. 12, no. 1, May 2022, Art. no. 7607, https://doi.org/10.1038/s41598-022-11684-w.

[20] M. Levitt and C. Chothia, "Structural patterns in globular proteins," *Nature*, vol. 261, no. 5561, pp. 552–558, Jun. 1976, https://doi.org/10.1038/261552a0.

[21] P. Kumar, S. Bankapur, and N. Patil, "An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features," *Applied Soft Computing*, vol. 86, Jan. 2020, Art. no. 105926, https://doi.org/10.1016/j.asoc.2019.105926.

[22] J. Selbig, T. Mevissen, and T. Lengauer, "Decision tree-based formation of consensus protein secondary structure prediction," *Bioinformatics*, vol. 15, no. 12, pp. 1039–1046, Dec. 1999, https://doi.org/10.1093/bioinformatics/15.12.1039.

[23] B. Yang, Q. Wu, Z. Ying, and H. Sui, "Predicting protein secondary structure using a mixed-modal SVM method in a compound pyramid model," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 304-313, Mar. 2011, https://doi.org/10.1016/j.knosys.2010.10.002.

[24] M. H. Zangooei and S. Jalili, "PSSP with dynamic weighted kernel fusion based on SVM-PHGS," *Knowledge-Based Systems*, vol. 27, pp. 424–442, Mar. 2012, https://doi.org/10.1016/j.knosys.2011.11.002.

[25] Z. Aydin, Y. Altunbasak, and M. Borodovsky, "Protein secondary structure prediction for a single-sequence using hidden semi-Markov models," *BMC Bioinformatics*, vol. 7, no. 1, Mar. 2006, Art. no. 178, https://doi.org/10.1186/1471-2105-7-178.

[26] J. Martin, J.-F. Gibrat, and F. Rodolphe, "Analysis of an optimal hidden Markov model for secondary structure prediction," *BMC Structural Biology*, vol. 6, no. 1, Dec. 2006, Art. no. 25, https://doi.org/10.1186/1472-6807-6-25.

[27] W. Yang, K. Wang, and W. Zuo, "Prediction of protein secondary structure using large margin nearest neighbour classification," *International Journal of Bioinformatics Research and Applications*, vol. 9, no. 2, pp. 207–219, Jan. 2013, https://doi.org/10.1504/IJBRA.2013.052445.

[28] A. Drozdetskiy, C. Cole, J. Procter, and G. J. Barton, "JPred4: a protein secondary structure prediction server," *Nucleic Acids Research*, vol. 43, no. W1, pp. W389–W394, Jul. 2015, https://doi.org/10.1093/nar/gkv332.

[29] D. W. A. Buchan, S. M. Ward, A. E. Lobley, T. C. O. Nugent, K. Bryson, and D. T. Jones, "Protein annotation and modelling servers at University College London," *Nucleic Acids Research*, vol. 38, no. suppl_2, pp. W563–W568, Jul. 2010, https://doi.org/10.1093/nar/gkq427.

[30] Z. Li and Y. Yu, "Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks." arXiv, Apr. 25, 2016, https://doi.org/10.48550/arXiv.1604.07176.

[31] A. Busia and N. Jaitly, "Next-Step Conditioned Deep Convolutional Neural Networks Improve Protein Secondary Structure Prediction." arXiv, Feb. 13, 2017, https://doi.org/10.48550/arXiv.1702.03865.

[32] R. Heffernan, Y. Yang, K. Paliwal, and Y. Zhou, "Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility,"

Bioinformatics, vol. 33, no. 18, pp. 2842–2849, Sep. 2017, https://doi.org/10.1093/bioinformatics/btx218.

[33] Y. Guo, W. Li, B. Wang, H. Liu, and D. Zhou, "DeepACLSTM: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction," *BMC Bioinformatics*, vol. 20, no. 1, Jun. 2019, Art. no. 341, https://doi.org/10.1186/s12859-019-2940-0.

[34] C. Fang, Y. Shang, and D. Xu, "MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, no. 5, pp. 592–598, 2018, https://doi.org/10.1002/prot.25487.

[35] M. R. Uddin, S. Mahbub, M. S. Rahman, and M. S. Bayzid, "SAINT: self-attention augmented inception-inside-inception network improves protein secondary structure prediction," *Bioinformatics*, vol. 36, no. 17, pp. 4599–4608, Nov. 2020, https://doi.org/10.1093/bioinformatics/btaa531.

[36] J. Hanson, K. Paliwal, T. Litfin, Y. Yang, and Y. Zhou, "Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks," *Bioinformatics*, vol. 35, no. 14, pp. 2403–2410, Jul. 2019, https://doi.org/10.1093/bioinformatics/bty1006.

[37] M. S. Klausen *et al.*, "NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 6, pp. 520–527, 2019, https://doi.org/10.1002/prot.25674.

[38] Uzma, U. Manzoor, and Z. Halim, "Protein encoder: An autoencoder-based ensemble feature selection scheme to predict protein secondary structure," *Expert Systems with Applications*, vol. 213, Mar. 2023, Art. no. 119081, https://doi.org/10.1016/j.eswa.2022.119081.

[39] A. Elnaggar *et al.*, "Ankh ☥ : Optimized Protein Language Model Unlocks General-Purpose Modelling." bioRxiv, Jan. 18, 2023, https://doi.org/10.1101/2023.01.16.524265.

[40] T. S. Mian, "Evaluation of Stock Closing Prices using Transformer Learning," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11635–11642, Oct. 2023, https://doi.org/10.48084/etasr.6017.

[41] M. Steinegger and J. Soding, "Clustering huge protein sequence sets in linear time," *Nature Communications*, vol. 9, no. 1, Jun. 2018, Art. no. 2542, https://doi.org/10.1038/s41467-018-04964-5.

[42] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, and C. H. Wu, "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, Mar. 2015, https://doi.org/10.1093/bioinformatics/btu739.

[43] A. Vaswani *et al.*, "Attention is All you Need," in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, Dec. 2017, vol. 30, pp. 1–15.

[44] A. Zemla, C. Venclovas, K. Fidelis, and B. Rost, "A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment," *Proteins: Structure, Function, and Bioinformatics*, vol. 34, no. 2, pp. 220–223, 1999, https://doi.org/10.1002/(SICI)1097-0134(19990201)34:2<220::AID-PROT7>3.0.CO;2-K.

[45] I. Drori *et al.*, "High Quality Prediction of Protein Q8 Secondary Structure by Diverse Neural Network Architectures." arXiv, Nov. 17, 2018, https://doi.org/10.48550/arXiv.1811.07143.