National College of Ireland

Eight-state Protein Secondary Structure Prediction Using NLP and Deep Learning

MSc Research Project
MSc Data Analytics

Anjali Augestin
Student ID: x23155086

School of Computing
National College of Ireland

Supervisor: Hamilton Niculescu

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anjali Augestin
Student ID:	x23155086
Programme:	MSc Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Hamilton Niculescu
Submission Due Date:	12/12/2024
Project Title:	Eight-state Protein Secondary Structure Prediction Using NLP and Deep Learning
Word Count:	7341
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Anjali Augestin
Date:	29th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Abbreviations

- PSSP: Protein Secondary Structure Prediction
- NLP: Natural Language Processing
- DSSP: Dictionary of Secondary Structure of Proteins
- LSTM: Long Short-Term Memory
- PSSM: Position Specific Scoring Matrix
- Glove: Global Vectors for Word Representation
- ESM: Evolutionary Scale Modeling
- GSN: Generative Stochastic Network
- CNN: Convolutional Neural Network

Eight-state Protein Secondary Structure Prediction Using NLP and Deep Learning

Anjali Augestin
x23155086

Abstract

The protein secondary structure prediction(PSSP) is a significant task in bioinformatics, as it determines the structural arrangement such as -helices, beta-sheets, and random coils, of amino acids. These structures are used to identify the 3D structure of protein, which in turn determines the function of each protein. This research mainly investigates the effects of Natural Language Processing (NLP) techniques in integration with deep learning models to predict the eight-state protein secondary structure prediction. NLP methods such as Word2Vec, GloVe, and ESM are used for retrieving embeddings from the amino acid sequences and the study compares their effectiveness in capturing contextual protein features.The LSTM and BiLSTM with attention mechanisms used for model training, improve prediction accuracy, while challenges such as class imbalance and the inability to identify all DSSP8 states remain. The findings highlight the potential of language models but emphasize the need for incorporating additional features like PSSM and resampling strategies to enhance class prediction. This study lays a foundation for future work in integrating contextual information for improved PSSP accuracy.

***Keywords*— Bioinformatics, protein secondary structure prediction, Natural Language Processing, Word2vec, glove, ESM, LSTM, BiLSTM.**

1 Introduction

Proteins play a vital role in all biological processes, in which the structures of each protein are related to their functions. The amino acid sequence is the primary structure of protein, while the protein secondary structure describes the folding patterns of these amino acids, which contribute to forming the stable tertiary structure. Among these, secondary structure prediction is particularly important as it serves as a foundation for determining tertiary structures and advancing fields like drug discovery and protein engineering (Srushti et al., 2023). Initially, the protein structures are classified into helix, coil, and strand. Later, the protein is classified into a more detailed eight states: G (helix), I (π-helix), H (α-helix), B (β-bridge), E (β-sheet), T (turn), S (bend or high curvature loop) and C (coil), known as DSSP8 labels (Kabsch and Sander, 1983). The research focuses on the prediction of these eight states, known as the Q8 problem.

Predictive models are commonly utilized for predicting the eight-state classes, but not focusing on the extraction of contextual meaning from the amino acid sequence itself but not focusing on the extraction of contextual meaning from the amino acid sequence. Recent progress in deep learning and NLP has improved protein secondary structure prediction. These methods treat protein sequences like language, making it easier to analyze their patterns. Each

amino acid can be seen as a word, and a protein sequence can be understood as a sentence, forming a linguistic representation of the sequence. This perspective allows NLP techniques, originally developed for text, to model protein sequences effectively (Jha et al., 2023). Based on this, researchers have developed protein-specific Language Models like SeqVec (Heinzinger et al., 2019) and ProtTrans (Elnaggar et al., 2020), to extract the vector representations of protein. While these advancements have had a significant impact on bio-informatics and biotechnology, they come with challenges such as high computational power and runtime, which limit their scalability and applicability to large datasets (Høie et al., 2022).

The success of language models (LMs) in natural language processing (NLP) has inspired this research to explore the effects of basic embeddings methods, inorder to address the computational expense of the above models. Additionally, the research addresses the gap of an efficient NLP techniques which requires less resource and runtime for the secondary structure prediction of protein. Thus, this research mainly utilizes NLP methods - Word2Vec, Glove and ESM, which are computationally less expensive. By utilizing these less expensive methods, medical practitioners can achieve faster results for time-sensitive protein-related tasks while also reducing costs in the medical field. Efficient methods helps in faster analysis of protein structure which in turn enables quicker diagnosis of genetic disorders, identifying pathogenic proteins, and developing treatments for diseases. Moreover, computationally expensive models require costly hardware or cloud services, whereas less resource-intensive methods allow hospitals, clinics, and research labs with limited access to high-performance computing resources to leverage these methods for protein analysis.

Research Question:

"How is the prediction of eight-state protein secondary structure influenced by deep learning models combined with NLP techniques for sequence analysis and feature extraction?"

1.1 Research Objectives

- To explore the improvements of the deep learning models integrated with NLP for the prediction of secondary structure of protein .
- To explore the efficiency of NLP techniques to extract features from proteins while minimizing computational resources and time.
- To extract vector representations of amino acid sequence and analyze amino acid sequence using NLP embedding extraction techniques.
- To identify the advantages and disadvantages of the NLP integrated system compared to the deep learning models.
- To compare and assess which NLP embedding extraction technique yields the best results for protein prediction.

The report comprise comprises following sections: Section 2 discussed the review of related works, section 3 is research methodology, which includes the methods and frameworks used in the research. Section 4 describes how the investigation is implemented, which is followed by the section 5 - evaluation and results. Final section concludes and discusses the future improvements of the work.

2 Related Work

Protein structure prediction is always been a crucial task in bio-informatics and advancements are still conducting in this particular area. The section discusses the techniques and limitations of the related works and indicates how it motivated and assisted in this research. The sections has four subsections in which first part reviews the datasets used, followed by a comparison of models utilized. Then third section involves the details about language models on protein, followed by the gap and the solution. Finally, the section is concluded in the summary.

2.1 A Review on Datasets and Methods Used in Protein Structure Prediction

The datasets used and methods employed in the prediction can create a great impact on the accurate prediction. The accuracy and performance of predictions may vary depending on the datasets, scoring matrices, and structure assignment algorithms utilized. Lin et al. (2016), utilized 4prot and CullPDB, which are large datasets. The 4prot dataset is splitted into train, validation and test set. The model is trained using the train set, fine-tuned using validation set performance, and the final results evaluated on the test set. The CullPDB dataset was selected, ensuring that sequences with over 25% identity to the CB513 dataset were excluded. The training and validation sets were derived from CullPDB, while the CB513 dataset was used as the test set to enable comparison with previous studies. The above study implies the effect of CullPDB and CB513 datasets on the model implementation.

The widely used dataset server for the eight state prediction is CullPDB as highlighted by Srushti et al. (2023). For training and testing, the dataset containing 5926 sequneces - "CullPDB5926" is utilized . The "CullPDB5926filtered" dataset from the CullPDB server is processed and the result is tested aginst the CB513 dataset. By this approach, the research acquired a better result and enabled them to support the performance of the model to unseen data as well. PSSM and one-hot vectors are used as the input features for model.Additionally, the research enhanced the model performance by using an ensemble model which includes two different deep learning models. The approach of training on one data and testing on another unseen data inspired and utilized for this research to analyze the ability of model to predict on new data.

Sofi and Wani (2022), utilized publicly accessible datasets, including CB6133, CB513, CASP10, and CASP11. The CB6133 dataset is a non-redundant collection of 6128 proteins derived from CullPDB by Wang and Drunbrack (2002). Additionally, the publicly available benchmark dataset CB513 is used by them exclusively for testing, while CASP10 and CASP11 are employed for model evaluation, containing 123 and 105 protein sequences, respectively. The input features for this research include protein coding features (21 dimensions), PSSM, conservation score (1 dimension), and seven physico-chemical properties of amino acids. PSSM profiles were generated using the PSI-BLAST method, with an e-value threshold of 0.001 and three iterations, and homologous sequences were retrieved from the UniRef90 database. The eight-state secondary structures were assigned using the DSSP labels, and the seven physico-chemical properties of amino acids were obtained from Meiler's study (2002).

Based on the literature survey, the CullPDB and CB513 datasets are frequently used in research and have consistently got superior results. Therefore, these datasets have been chosen for my investigation as well.

2.2 A Comparison of Applied Models on Protein Secondary Structure Prediction

The comparison of dataset used and algorithm applied for the protein secondary structure prediction is shown in table 1. Various datasets and deep learning models are widely used for the prediction of protein secondary structure, which are listed in below table. Among the deep learning methods, the models trained using LSTM method acquired the highest accuracy compared to other machine learning methods. Thus, it is evident from the below table that, LSTM technique is efficient in capturing long-range dependencies and handling variable-length sequences.

Paper	**Datasets**	**Applied Technique**	**Results(accuracy)**
Jin et al. (2021)	CASP10, CASP11, CASP12, CB513 and TS115	GCN and BiLSTM	CASP10-78.05, CASP11-76.81, CASP12-72.84, CB513-74.46, TS115-76.04
Zeng et al. (2022)	CASP14set, TEST524	BiLSTM with Bootstrap Aggregating	CASP14-69.95, TEST524 -65.61
Wang et al. (2019)	CB513	Ensemble of LSTM Neural Networks	77.9%
Rahman et al. (2023)	ccPDB 2.0	LSTM and BiLSTM	Lstm-83.24, BiLstm-89.10
DeepCNF Wang et al. (2016)	CASP14set	DeepCNF	63.85

Table 1: comparison of applied models on protein secondary structure prediction

2.3 An Analysis on Language Models on Protein

NLP methods have been widely used in bioinformatics in various fields of protein prediction. Researchers have conducted experiments in building language models for addressing different protein tasks. The two significant models are ProtTrans (Elnaggar et al., 2020) and Seqvec (Heinzinger et al., 2019), developed on computationally huge deep learning models like transformers and trained on billions of protein data. The ProtTrans is trained on more than 300 billion amino acids and developed on Transformer-XL, XLNet, BERT, Albert, Electra and T5. But the model is not actually used for the Q8 prediction, it is only validated for the three-state prediction and other tasks. Seqvec (Heinzinger et al., 2019), bidirectional LSTM-based architecture trained on the large UniRef50 dataset. However, this model finds difficult to achieve good results(68%) for the Q8 prediction compared to other researchers mentioned in the above comparison.

The effect of language models on protein are also discussed in research conducted byJha et al. (2023). They extensively used the NLP embedding technique for feature extraction. The proposed approach considers two representations of a protein: the amino acid sequence and the 3D structure. They employ a language model and a vision transformer model, leveraging transfer learning to extract feature vectors from these respective protein representations. The SeqVec embedding method is utilized to extract the embedding and contextualized value of the amino acid sequence.

The vast possibilities of NLP in the protein predictions are discussed by Ofer et al. (2021). The paper features different types of NLP techniques that can create a great impact for the

predictions. They discuss the concept of language of proteins and, framework of considering and treating amino acid sequences as a sentence. The paper investigates diverse approaches for the purpose of protein sequence encoding as text and utilizing Natural Language Processing (NLP) techniques for analysis. It covers traditional methods like bag-of-words as well as recently developed approaches like word and embeddings, language models on protein sequence. The possibilities to apply the word embeddings techniques for protein structure prediction is emphasized and explained the effect of such NLP techniques in the prediction.

2.4 Gap Identified and Investigating Solution

Protein secondary structure prediction is a vital field of research, with eight-state secondary structure prediction posing significant challenges. Chen et al. (2016) emphasized the importance of feature extraction by employing a support vector machine (SVM) classifier to utilize essential protein sequence features. Their approach integrated PSSM some specific features of protein and given as input to the model, which ensured the finding of homologous protein structures. Additionally, they combined Hydrophobicity Sequence Features (HSF) with sequence to identify which attributes most accurately predicted protein structures.They used the 12 HSF features and 10 structural features of protein for model optimization, which assisted them to conclude that structural features were vital for protein structure prediction, effectively capturing sequence conformation. In contrast,they also identified that hydrophobic features were less significant in the prediction process. They focused on predicting the structure using the properties of protein rather than analyzing and extracting the features from the amino acid strings.

A similar study was conducted by Sofi and Wani (2022) which utilized the physical properties of protein as features. Their study utilized diverse protein features, including 21-dimensional protein coding features and PSSM, and examined the impact of conservation scores on structure prediction. They aimed to improve prediction accuracy by integrating physical properties into classification models. The researchers proposed a deep learning approach combining two deep learning models- CNN and LSTM. Additionally, they included an attention mechanism to capture features effectively along with the model. Their experiments, conducted on four datasets -CB6133, CB513, CASP10, and CASP11, involved two setups. The first used PSSM along with sequence features, and used seven amino acid properties for the second one. Results showed that the first experiment performed well across datasets, while the second demonstrated improved accuracy, highlighting the significant role of physical properties in structure prediction. This study also, lacks the proper feature extraction from the amino acid strings, which is the major aspect that addressed in my investigation.

But, Zhou and Troyanskaya (2014) focused on enhancing prediction accuracy by extracting prominent features from amino acid sequences and highlights the difficulty of achieving balanced accuracy across all eight classes. Specifically, predicting S (bend) and G (310-helix) classes proved challenging due to severe class imbalance and insufficient training data. Additionally, the rare class I presented significant difficulties, with no predictions made for this category. Rather than addressing the imbalanced nature of the dataset, their research focused on extracting significant features from amino acid sequences. A generative stochastic network (GSN) and a CNN network is proposed, to capture features for the prediction.

Collectively, these studies demonstrate different ways the feature engineering applied in extraction in secondary structure prediction of protein. Chen et al. (2016) and Sofi and Wani (2022) utilized properties of protein to improve prediction accuracy, while Zhou and Troyanskaya (2014) focused on extracting features from sequences itself using GSN, but faced some limitations. A detailed review of these studies reveals a gap in proper extraction of features from amino acid strings itself rather than using physical properties of amino acids. So the research aims to investigate about effective methods for the extraction of features from amino acid string. Moreover, these findings emphasize that selecting and applying features thoughtfully

can significantly boost model performance and classification accuracy, highlighting the crucial role of feature extraction techniques in this field.

2.5 Summary

The literature survey primarily highlights two key challenges in protein structure prediction and emphasizes the need for more efficient and computationally cost effective amino acid feature extraction techniques that extracts contextual information from the amino acids to enhance protein structure prediction. First, as discussed in section 2.3, the current language models (Elnaggar et al., 2020), (Heinzinger et al., 2019) are not only highly computational expensive but also finds difficult to achieve better results for secondary structure prediction, but works well for other protein tasks. Also the researches mentioned in section 2.4, indicates the need for the extraction of contextual information from the amino acid string itself.
Therefore combining these two problems, this research introduces an approach to investigate the impact of basic NLP techniques on protein structure prediction. The study focuses on utilizing NLP-based feature engineering methods to extract word embeddings from amino acid sequences which requires less computational resource and time. The widely used CullPDB and CB513 datasets were selected to ensure robust evaluation of the embedding methods. For modelling, an LSTM model was chosen due to its superior ability to handle sequence-related tasks effectively.

3 Research Methodology

Data Mining process constitutes of various techniques like CRISP-DM, SEMMA and KDD. The KDD-Knowledge Discovery in Databases methodology is selected as the research methodology among the other methodologies, which is shown in Fig 1. The systematic approach in KDD to extract valuable insights from data is helpful in identifying the protein sequence features correctly. Moreover, protein structure prediction requires extensive data preprocessing and correct feature engineering, KDD emphasizes the importance of data cleaning and preparation, ensuring the input data is consistent and reliable. The investigation is implemented in python language utilizing many python libraries in each phase of the investigation.

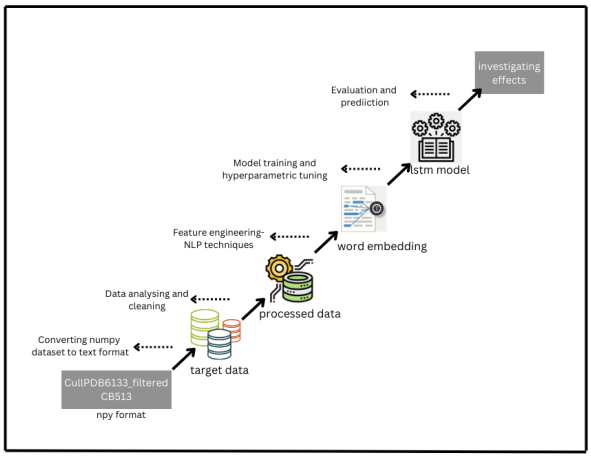

Figure 1: Research methodology for eight-state protein structure prediction

The process of eight secondary structure prediction of protein is an iterative process, which

involves the conversion of raw data into processed data according to the prediction requirements. The following steps and frameworks are used for the investigation:

3.1 Data collection and Dataset preparation

The selection and preparation of appropriate data are fundamental steps in any research study. For this investigation on protein secondary structure prediction, two datasets were utilized: CullPDB6133_filtered and CB513, which were originally extracted from Zhou and Troyanskaya (2014), which can accessed through the link: `https://zenodo.org/records/7764556#ZByi1ezMJvI`. The model is trained and validated on the CullPDB6133_filtered dataset and tested on the benchmark dataset CB513. The CB513 dataset is frequently used to evaluate and compare the effectiveness of protein secondary structure prediction techniques because CullPDB6133filtered dataset and CB513 are disjoint (Srushti et al., 2023), thus it is used as the test set in this investigation.
The datasets were downloaded and processed to convert them into a human-readable format suitable for analysis, using the algorithm given in its original source .After processing, the datasets were organized into meaningful components for the research. Specifically:

1. **pss_train'**: Contains the encoded secondary structure labels (DSSP8 categories) from CullPDB6133_filtered, for the training set.

2. **'pss_test'**: Contains the encoded secondary structure labels (DSSP8 categories) from CB513, for the test set.

3. **'aminoacid_train'**: Consists of amino acid sequences from CullPDB6133_filtered, used for training the model.

4. **'aminoacid_test'**: Consists of amino acid sequences from CB513, used for evaluating the model.

This structured organization of data facilitates efficient training, validation, and testing of the models used in this study.

3.2 Data analyzing and preprocessing

The extracted data is then analyzed and visualized for understanding the data structure and data types. The data is analyzed using the powerful and widely-used library in Python - 'pandas'. Also, the data is visualized for the better understanding of the dataset structure .The distribution of target variable, distribution of sequence lengths and comparison of sequence lengths and DSSP8 labels are visualized using the python libraries 'seaborn' and 'matplotlib'. Afterwards, the data is cleaned to avoid missing data and for removing white spaces from the data.
The target variables are converted into numerical format using the 'LabelEncoder' from scikit-learn python library, which ensures consistent and efficient model training.

3.3 Feature Engineering

This is the most crucial process in this research, where the test and train amino acid data is transformed into embeddings. The embeddings are extracted from the amino acid strings using three NLP techniques- Word2Vec, Glove and ESM. Additionally, another significant transformation is done by label encoding the protein secondary structure labels.

NLP TECHNIQUES

Natural language processing (NLP) is a field of computer science which enables machine to understand human language. Proteins, which can be represented as strings of amino-acid letters, are a natural fit to many NLP methods (Ofer et al., 2021). Thus, the NLP feature extraction techniques are used to extract the embedding from the amino acid string in this investigation. The emebdding methods presented in Ofer et al. (2021) for encoding the information of proteins as text and analyzing it with NLP methods are considered for the investigation. Three embedding methods are mainly considered:

(a) **Word2vec:** Word2vec is one of the method in NLP for extracting the vector representation of word.These vectors carries information about the word based on the surrounding words on their co-occurrence patterns in the sequences (Ofer et al., 2021), which is assumed to assist to capture the information of each amino acid according to their position and surrounding tokens. Therefore, this model is opted as one of the NLP technique to investigate the effects. In this research, the Word2vec embedding extraction is enabled by using 'gensim' library in python.

(b) **GloVe:** Glove is another NLP method similar to Word2Vec, but it captures the global context rather than local contextual information.The method was originally developed by Pennington et al. (2014) in Stanford University. The "glove.6B.100d.txt" word vector representation is downloaded from its official site `https://nlp.stanford.edu/projects/glove/`. The specific embedding file, "glove.6B.100d.txt", was selected as it represents each word with a 100-dimensional vector. The dimension is chosen as 100 because, it maintains a balance between preserving sufficient information and maintaining computational efficiency, making it particularly suitable for protein related prediction tasks using GloVe embeddings (Bepler and Berger, 2019). The method is chosen to investigate the effect of the Glove embeddings for the protein structure prediction task.

(c) **Evolutionary Scaling Model(ESM):** The ESM, specifically designed for protein sequences, trained on large-scale biological data. The model was developed by Meta Fundamental AI Research Protein Team (FAIR) for the accurate protein structure and alphafold predictions, trained on 250 million sequences of the UniParc database , which has 86 billion amino acids (Ismi et al., 2022). The latest version, ESM2 is used as the third NLP method in this investigation as it outperforms all tested single-sequence protein language models across a range of structure prediction tasks (Lin et al., 2022).This model was implemented using the 'fair-esm' Python library, which provides seamless access to pretrained ESM models for protein analysis.

3.4 Model Training and Hyperparametric Tuning

In this phase, the extracted embeddings from all the three methods are used to train an LSTM model. All three models are implemented through 'tensorflow' library, as it the most and efficient and widely used for deep learning models for protein structure prediction (Srushti et al., 2023), (Ghosh and Shill, 2021). Key functionalities from the TensorFlow Keras module, such as LSTM, Dense, Embedding, TimeDistributed, Bidirectional, and Dropout, are employed to construct a robust architecture for training.

The model is fine-tuned to evaluate the impact of the applied NLP-based embedding techniques on predicting protein secondary structure. Hyperparameters such as learning rate and the number of epochs are optimized to ensure improved performance. Additionally, an attention mechanism is incorporated into the model to enhance its ability to focus on critical aspects of the sequence data. This mechanism effectively addresses the challenges posed by low-frequency classes within the DSSP8 labels and prioritizes relevant sequence regions, thus improving prediction accuracy (Sofi and Wani, 2022), (Mohamed Mufassirin et al., 2023).

3.5 Evaluation and Knowledge Discovery

Finally, all the three experiments are evaluated using the overall accuracy, and per class precision, f1-score and recall as evaluation metrics. The modules 'confusion_matrix' and 'classification_report' are used for retrieving the confusion matrix and classification report.Then, the influence of the three applied techniques on the protein secondary structure prediction is compared.

4 Design Specification

The design process of this investigation includes three layers – data layer, processing layer and presentation layer. The data extracted was in numpy format and it need to be converted to human readable text format for the further processing. Hence, 3-tier architecture is followed for this investigation.

The data layer includes the collection, storage and preparation of the data. The raw data downloaded is processed and converted into text format. The processing layer handles the core implementation processes like data preprocessing, feature extraction and model training. In the presentation layer, the outcomes of the processing layer are showcased in an interpretable and user-friendly manner. The presentation layer includes the graphs, classification reports and comparison of different NLP techniques and their performance.
The fig 2, depicts design architecture for the eight-state protein secondary structure prediction.

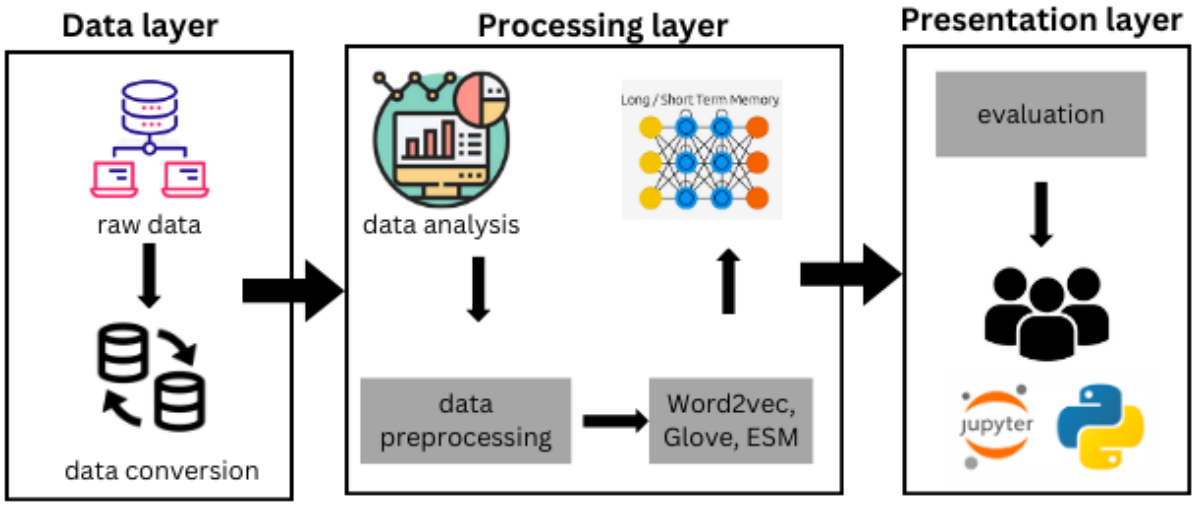

Figure 2: The three-tier architecture for eight-state secondary structure prediction

5 Implementation

5.1 Data Analysis

The data is analysed as part of the initial data exploration. The first few rows of test and train sets are displayed using 'head()' function to understand their structure, content, and format. It helped to verify that the data was loaded correctly. Then, the shape of the training and testing sets are retrieved and found that the training set has 5534 and the test set has 514 rows.
Missing values can negatively impact model performance, so the dataset is checked for missing. The dataset found to be clean without any missing values.

5.2 Exploratory Data Analysis

Various visualizations have been done in order to analyze and explore the dataset. For this investigation, the visualizations include distribution of DSSP8 labels, the variability in protein sequence lengths, and the relationship between sequence lengths and their corresponding DSSP8 labels. The potential issues including the class imbalance and inconsistency in datasets can be identified and addresses by comprehending these visualizations.

The bar plot visualization of the frequency distribution of DSSP8 secondary structure labels across the dataset is shown in Fig 3. The plot indicates that the DSSP8 labels are imbalanced and the classes such as H and I are rare compared to other classes. The class B is present in high frequency in the corresponding secondary structure labels of protein. This imbalance poses a challenge for the model, as it risks overfitting to the dominant classes and performing poorly on minority class predictions.

Although common resampling techniques, such as oversampling or undersampling, could help mitigate this issue, they were not implemented in this investigation due to resource and time constraints. Instead, to enhance the focus on underrepresented classes, an attention mechanism was incorporated during hyperparameteric tuning, enables the model to emphasize key sequence regions.

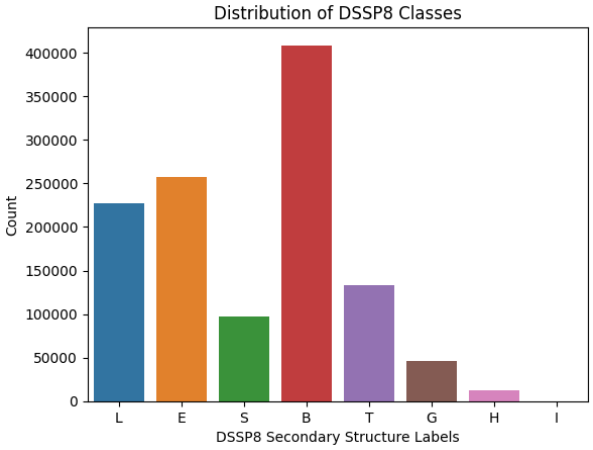

Figure 3: Distribution of DSSP8 labels

The variability in the lengths of protein sequences is observed by plotting a histogram, which is shown in Fig 4. The histogram reveals a peak within the 500-600 range, indicating that the most common protein sequence lengths fall within this interval. Furthermore, the broad range of the plot suggests that the dataset includes a diverse array of sequence lengths, including both short and long protein sequences. The inclusion of both short and long sequences ensures that the model is exposed to varied structural contexts, enhancing its ability to generalize across different protein types and sizes.

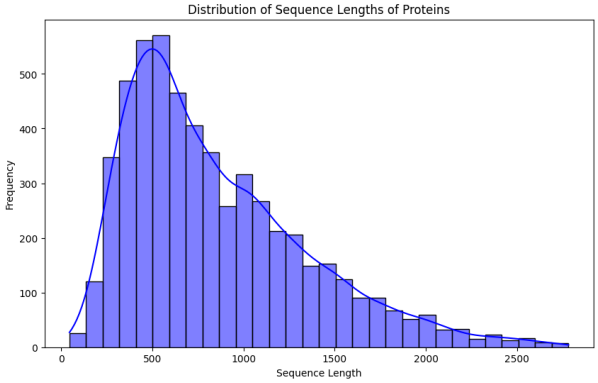

Figure 4: Distribution of sequence lengths of protein

Finally, a scatter plot is plotted to compare the lengths of protein sequences and their corresponding DSSP8 labels to check for consistency. The plot is illustrated in Fig 5, in which the diagonal line indicates a one-to-one correspondence between sequence length and DSSP8 label length, signifying that each amino acid in the sequence has a corresponding secondary structure label. Additionally, the plot confirms the absence of outliers in the dataset. This consistent alignment between sequence and label lengths ensures data integrity, which is essential for the effectiveness of supervised learning models.

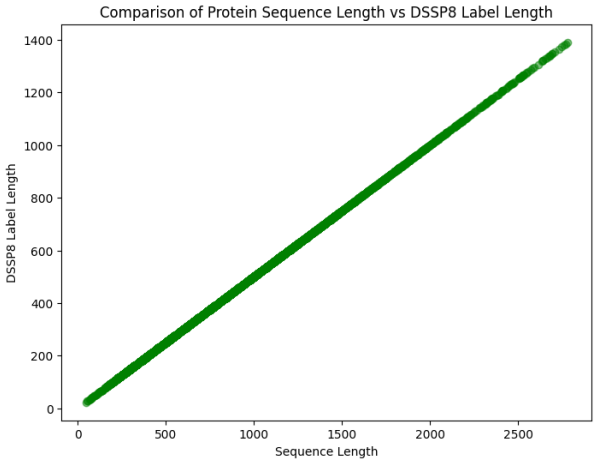

Figure 5: Comparison of protein sequence length and DSSP8 labels length

5.3 Data Preprocessing

The raw data extracted is converted into a structured format for making the data suitable for feature engineering and deep learning models. The data processing steps include the sequence tokenization, white space removal and label encoding.

Sequence Tokenization: In this step, each amino acid sequence is converted into individual characters using 'tolist()' function in python, where token represent an amino acid. The amino acid sequence is converted into a list of amino acid tokens, preparing the the dataset for embedding extraction and model training.

Data cleaning: When the data is tokenized the white spaces are also considered and tokenized. Therefore, the white spaces are removed from the list for the proper extraction of embedding without the whitespace.

Label encoding The target variables, DSSP8 labels are label encoded using the 'labelencoder()' function in 'Scikit-learn' library. The individual tokens are converted into numerical format required for model training. The conversion ensures that encoding is consistent across datasets as well as prevent the model from potential mismatch errors during testing.
One-hot encoding: The 'to_categorical()' function from 'Keras' on tensorflow is employed for this conversion. The numeric labels are converted into one-hot encoded vectors, where each label is represented as a binary vector with a 1 in the index corresponding to its class and 0 elsewhere. The one-hot conversion ensures the consistent input for the deep learning model.

5.4 Embedding Methods - Feature Engineering

This is one of the most significant phase in this research where the embeddings are extracted from the amino acid strings using three different NLP techniques. These embeddings are then given as the input to the LSTM model. The investigation is done as three different experiments, each experiment includes the embedding extraction using the different NLP methods and model training using LSTM.

5.4.1 Embedding Extraction Using Word2vec

The Word2Vec model was trained on the tokenized amino acid sequences from the training dataset using the Word2Vec implementation from the gensim library. The parameters defined for training - A vector size of 50 was selected, as 50-dimensional embeddings are widely used for compact and effective representations (Sivakumar et al., 2020). This choice ensures efficient mapping of each amino acid to a meaningful 50-dimensional vector.

Minimum Count (min_count) defined to 1, which ensures that even rare amino acids are included in the vocabulary.Finally, Workers parameter utilized 4 parallel threads for efficient training.

Embedding Extraction: A custom-built function named 'sequence_to_embedding' was developed as part of this research to transform each amino acid sequence into a corresponding series of embeddings The corresponding vector representation for each amino acid was retrieved from the trained Word2Vec vocabulary.

Padding for Consistency: The sequences of varying length from the embeddings are padded to ensure uniformity in the input for the LSTM model. The length of the longest sequence across the training and testing datasets was to set a standard size 'max_len'. For padding, all shorter sequences were padded with zero vectors at the end to match the maximum length, ensured fixed-length sequences for batch processing, which is a requirement for LSTM model. The resultant dataset is a padded matrix of size (number_of_sequences, max_len, 50) for training and testing set.

5.4.2 Embedding extraction using GloVe

Pre-trained GloVe embedding 'glove.6B.100d.txt' was downloaded and used, where each character or residue was represented by a 100-dimensional vector (embedding_dim = 100). The GloVe embeddings file was read line-by-line for loading the embeddings, each character and its corresponding vector were stored in a dictionary.

Embedding Matrix: An embedding matrix of size (vocab_size, embedding_dim) was created to map each amino acid token to its pre-trained vector.The embedding foe each amino acid was retrieved from the GloVe dictionary.If an amino acid was not present in the GloVe

embeddings, its row in the matrix was initialized to zeros. Finally, a padding matrix of size (number_of_sequences, 700) was generated for the training and testing sets.

5.4.3 Embedding extraction using ESM

The latest version of pre-Trained ESM model, 'ESM2-t6-8M_UR50D' is loaded and used for generating embeddings based on evolutionary relationships, learned from a massive protein database. The 'batch_converter' is also used to convert protein sequences into the structure required by the ESM model. It also comprehends and handles special tokens like start and end tokens, which are used to indicate the start and end of sequences.

Extracting ESM embeddings A function 'get_esm_embeddings' was created particularly to generate embeddings for a list of sequences. The train and test sets are passed into the function to retrieve the embeddings. The extracted embeddings capture both local and global context. The end and tokens are excluded to keep the focus only to amino acid strings. For obtaining a fixed-length vector representation for each sequence, the embeddings for all residues in a sequence are averaged.
The processes are repeated as batches for all sequences in the dataset and resultant dataset is converted into a matrix of shape (number_of_sequences, embedding_dim).

5.5 LSTM model for eight-state protein structure prediction

LSTM model is used for model training for all three experiments. The extracted embeddings are given to the defined LSTM model. The model is complied using 'Adam' optimizer and accuracy is used as the evaluation metrics during training and testing.

The model architecture includes then following layers:

Embedding Layer: The input to the model is given in the embedding layer, which is a sequence of protein embeddings with a fixed length (max_len) and dimensionality (embedding_dim). This input layer depicts the the extracted embeddings using methods such as Word2Vec, GloVe, or ESM.
LSTM Layer: A LSTM layer of 128 units is defined after the embedding layer. This layer holds the temporal dependencies in the protein sequence data, making it ideal for sequence prediction tasks. The the output of the LSTM layer should be a sequence that matching the length of the input, which is ensured by setting the parameter return_sequences=True'.
TimeDistributed Dense Layer 1: To reduce the dimensionality of the LSTM output as well as to produce linear output, a fully connected layer is applied with 64 hidden units and a 'ReLU' activation function at each step, Using the ReLU activation function, the model achieves faster training compared to traditional activation functions like Sigmoid and Tanh (Ghosh and Shill, 2021).
TimeDistributed Dense Layer 2: Another dense layer with 'softmax' activation at each time step enables a valid probability distribution for all target classes. The Softmax function is usually used for the deep, learning classification models, because it ensures that output probability of the model Classification represents the probability that the input falls into each of the classes (Ghosh and Shill, 2021).
Output Layer: It outputs the sequence of predictions for the secondary structure of the input data.

Each model is compiled using 'Adam' optimizer, which is efficient for handling large datasets. The loss function is also defined as categorical crossentropy and accuracy is used as the evaluation metrics for all three experiments. The models are trained using the processed amino

acid strings that is the embeddings from Word2vec, GloVe and ESM. Each model was trained for 10 epochs with a batch size of 32. The same dataset was utilized for both validation and model evaluation.

5.6 BiLSTM with Attention Mechanism

An enhanced model of BiLSTM with attention mechanism, dropout and hyperparametric tuning is implemented to explore the effects as mentioned in research methodology section. This enhanced model constitutes Bidirectional LSTM(BiLSTM), attention mechanism, dropout regularization and early stopping. The model architecture includes:

Bidirectional LSTM layers: The first BiLSTM layer contains 128 units and the second BiLSTM layer adds a layer with 64 units which enables the model to capture more dependencies and for deeper representation learning.
Dropout Layers: The dropout layers are added to reduce overfitting. Dropout layers with a rate of 0.3 are added after each BiLSTM layer. The value of 0.3 keeps a balance between reducing overfitting and preserving ability to capture information from the sequence. Moreover, Srushti et al. (2023) highlights, that 0.3 is typically an effective range for dropout rates, as it provides a moderate level of regularization.
TimeDistributed Dense Layers: The first dense layer is given with 64 units with ReLU activation function which produces linearity. The next dense layer uses 32 units with 'ReLU' activation, which reduces the dimensionality of sequence representation for attention mechanism.
Output Layer: 'Softmax' activation function is applied to the final time dsitributed layer which produce the distribution of all classes. The target variable classes are given as the number of units in this output layer.

The model includes the attention mechanism through these layer not only to improve focus on relevant parts of sequence, but also to have higher priority to less frequent classes in the target. Additionally, regularization with Early Stopping mechanism with a patience level of 3, is added to the model for monitoring validation loss. It ensures tarining halts early, if there is no progress in loss for 3 consecutive epocs. The parameter 'restore_best_weights' is assigned as True ensures the model returns to the best-performing state which avoids overfitting. Then all three models is tuned for 20 epocs, based on the these enhancements.

6 Evaluation

The research involves three experiments which investigates the effect of three NLP techniques-Word2vec, Glove and ESM on protein secondary structure prediction.These methods are utilized to generate embeddings from amino acid sequences, which serve as input features for training the LSTM model. Each experiment investigates how these distinct embedding strategies influence the predictive performance of the model.

6.1 Experiment 1 - Word2vec with LSTM

The LSTM model initially achieved an overall accuracy of 89.48%, while the tuned model with an attention mechanism improved to 91.89%. The classification report of the model before and after hyper-parametric tuning is shown in Fig 6. It is evident from the report that, the untuned model performed well on the majority class (B) but showed significant disparities in predictions for other DSSP8 classes. Also the training and validation accuracy plots shown small spikes in the accuracy and loss curves, indicating that the model might occasionally overfit to specific batches or encounter harder to learn patterns within the training data. The enhanced model

with attention mechanism as mentioned in methodology section is implemented and the model is tuned in order to address these issues as well as to explore the effect of the enhanced model.

	precision	recall	f1-score
B	0.96	0.98	0.97
E	0.18	0.00	0.00
G	0.00	0.00	0.00
H	0.26	0.64	0.37
I	0.00	0.00	0.00
L	0.49	0.07	0.12
S	0.00	0.00	0.00
T	0.00	0.00	0.00
accuracy			0.89
macro avg	0.24	0.21	0.18
weighted avg	0.87	0.89	0.87

(a) classification report before tuning

	precision	recall	f1-score
B	1.00	1.00	1.00
E	0.35	0.02	0.03
G	0.00	0.00	0.00
H	0.32	0.96	0.48
I	0.00	0.00	0.00
L	0.60	0.05	0.10
S	0.00	0.00	0.00
T	0.00	0.00	0.00
accuracy			0.92
macro avg	0.28	0.25	0.20
weighted avg	0.92	0.92	0.90

(b) Classification report after tuning

Figure 6: The classification reports of the model before and after hyper-parametric tuning

The classification report after hyper-parametric tuning shown in 6b, indicates that the precision of classes B, E, H and L increased, but the unidentified classes remains same. The plots in Fig 7 display the training and validation accuracy (left) and loss (right) after hyper-parametric tuning. The training accuracy starts lower but rises quickly, leveling off around 90% after a few epochs. This demonstrates that the model learns effectively from the training data. The validation accuracy remains consistently high, around 92%, with minimal fluctuations after the initial epochs, which indicates that the model generalizes well to unseen data and overfitting is minimal, a positive outcome of tuning and techniques like dropout, early stopping, and attention.

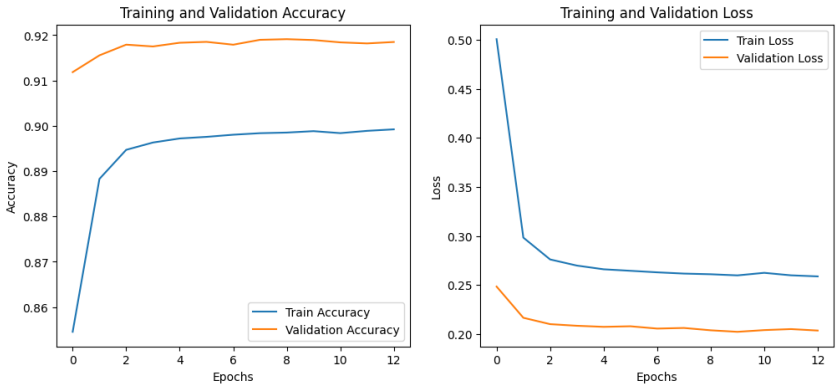

Figure 7: Accuracy and loss plots of Word2vec with LSTM after tuning

6.2 Experiment 2 - Glove with LSTM Model

The Glove embeddings is given as the input to the LSTM model trained for 10 epocs for the second experiment. The classification report of the model trained before and after hyper-parametric tuning is illustrated in Fig 8. Initially, the model achieved an overall accuracy of 77.1%, which improved to 83.62% after tuning. This model effectively identified the classes B, E, H and L similar to the first experiment but with lower precision for all identified classes

compared to the first experiment.The large gap between the weighted average precision (80%) and recall (77%) suggests overfitting to majority classes, particularly B.

	precision	recall	f1-score
B	0.95	0.92	0.94
E	0.33	0.00	0.00
G	0.00	0.00	0.00
H	0.24	0.83	0.37
I	0.00	0.00	0.00
L	0.62	0.05	0.10
S	0.00	0.00	0.00
T	0.00	0.00	0.00
accuracy			0.77
macro avg	0.27	0.23	0.18
weighted avg	0.80	0.77	0.75

(a) classification report before tuning

	precision	recall	f1-score
B	0.99	0.99	0.99
E	0.00	0.00	0.00
G	0.00	0.00	0.00
H	0.31	0.96	0.47
I	0.00	0.00	0.00
L	0.55	0.06	0.11
S	0.00	0.00	0.00
T	0.00	0.00	0.00
accuracy			0.84
macro avg	0.23	0.25	0.20
weighted avg	0.81	0.84	0.80

(b) Classification report after tuning

Figure 8: The classification reports of the Glove with LSTM model before and after hyper-parametric tuning

The classification report after tuning is shown in 8b. After tuning incorporated with attention mechanism the he weighted averages improved (precision: 81% and recall: 84%), indicating the model's better balance across classes, but still favoring the majority classes. Also, performance for class B improved significantly, with nearly perfect precision, recall, and F1-scores.Class H showed increment, with its F1-score increasing to 47%, indicating better recognition of this class. But, the class E identified before tuning, no longer identified at all, with zero recall and F1-scores due to attention mechanism may have overemphasized the majority classes like B and H. The the training and validation accuracy (left) and loss (right) after hyper-parametric tuning is shown in 9, which indicates the model performance without overfitting.

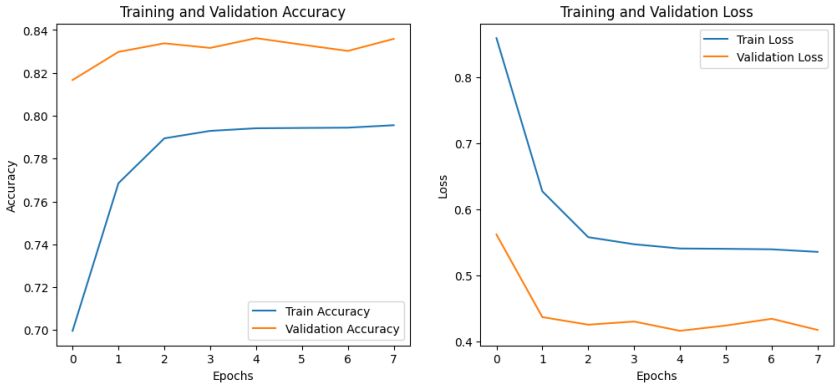

Figure 9: Accuracy and loss plots of Glove with LSTM after tuning

6.3 Experiment 3 - ESM with LSTM model

For the third experiment the ESM embeddings are given to LSTM model. Before tuning, the model acquired an accuracy of 53.63%, but only identifying the classes B with a precision of 76% and H with a lower precision of 24%. After hyper-parameter tuning, the model demonstrated improved performance by successfully identifying two additional classes, E with a precision of 21% and L with a precision of 53%. However, the precision for the majority class, B,

decreased to 71%, while the precision for class H remained nearly unchanged. This shift in performance suggests that the model, enhanced with an attention mechanism, adjusted its focus by prioritizing previously unidentified or underrepresented classes, reducing the bias toward the majority class.The classification report of the model before and after tuning is shown in fig 10, The validation-training accuracy and loss plots after tuning shown in fig 11, which indicates a the model performance with lower overfitting while faces challenges in maintaining consistent performance on the validation data.

Classification Report:

	precision	recall	f1-score
B	0.76	0.83	0.79
E	0.00	0.00	0.00
G	0.00	0.00	0.00
H	0.24	0.70	0.36
I	0.00	0.00	0.00
L	0.00	0.00	0.00
S	0.00	0.00	0.00
T	0.00	0.00	0.00
accuracy			0.54
macro avg	0.13	0.19	0.14
weighted avg	0.43	0.54	0.47

(a) classification report before tuning

Classification Report:

	precision	recall	f1-score
B	0.71	0.92	0.80
E	0.21	0.14	0.17
G	0.00	0.00	0.00
H	0.25	0.41	0.31
I	0.00	0.00	0.00
L	0.53	0.06	0.11
S	0.00	0.00	0.00
T	0.00	0.00	0.00
accuracy			0.56
macro avg	0.21	0.19	0.17
weighted avg	0.48	0.56	0.49

(b) Classification report after tuning

Figure 10: The classification reports of the ESM with LSTM model before and after hyper-parametric tuning

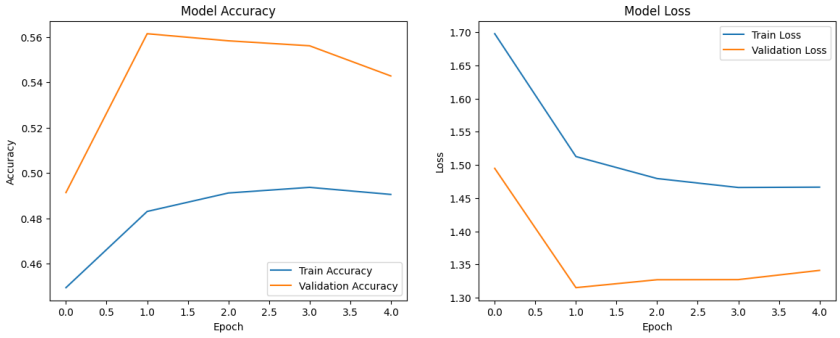

Figure 11: Accuracy and loss plots of ESM with LSTM after tuning

6.4 Discussion

The comparison of overall accuracy of three experiments is shown in table 2. The first experiment employed Word2Vec embeddings as input to the LSTM model. The model achieved 89.48% accuracy before hyperparameter tuning and improved to 91.89% after tuning. However, the issue of class imbalance affected the model evidently, as the model strongly favored the some classes such as B, E, H and L. In the second experiment with GloVe embeddings the model achieved 77.1% accuracy before tuning and 83.62% after tuning, showing a significant drop in accuracy compared to Word2Vec. Precision for all identified classes (B, E, H, and L) was notably lower. For the third experiment, utilising the ESM embeddings, LSTM model demonstrated improved focus on minority classes after hyperparameter tuning, particularly for class E and L.

Embedding method	**LSTM**	**BiLSTM with attention**
Word2Vec	89.48	91.89
Glove	77.1	83.62
ESM	53.63	56.14

Table 2: Comparison overall accuracy for three models

The comparison of per-class precision for all three models with the dropouts are depicted in table 3 for better understanding of the effects of both NLP techniques and hyper-parametric tuning applied. For the word2vec with LSTM model, the model identifies B, E, H and L classes before tuning, but after tuning the precision improved across all classes. But for the Glove with LSTM model, the model performance is dropped compared to word2Vec. Finally, for the ESM embeddings, the model identifies only B and H initially, and identifies two more classes, E and L after tuning.

Model	**Dropout**	**B**	**E**	**G**	**H**	**I**	**L**	**S**	**T**
LSTM + Word2vec	-	0.96	0.18	0.00	0.26	0.00	0.49	0.00	0.00
BiLSTM + Word2vec	0.3	1.00	0.35	0.00	0.32	0.00	0.60	0.00	0.00
LSTM + Glove	-	0.95	0.33	0.00	0.24	0.00	0.62	0.00	0.00
BiLSTM + Glove	0.3	0.99	0.00	0.00	0.31	0.00	0.55	0.00	0.00
LSTM + ESM	-	0.76	0.00	0.00	0.24	0.00	0.00	0.00	0.00
BiLSTM + ESM	0.3	0.71	0.21	0.00	0.25	0.00	0.53	0.00	0.00

Table 3: Precision of DSSP8 Classes for LSTM Models with Various Embeddings

The overall methods and architecture of the experiments was methodologically efficient, but the choice of embeddings and model configurations had a significant impact on the results. Static embeddings such as Word2Vec and GloVe demonstrated limitations capturing the complexity of protein secondary structures. In contrast, the ESM embeddings significantly enhanced performance, as the attention mechanism effectively shifted focus from the dominant classes to previously underrepresented or less accurately predicted classes, leading to improved balance in prediction. The hyperparameter tuning process proved beneficial, but further exploration of regularization techniques is needed for more robust model. The imbalance in the DSSP8 dataset continues to pose a significant challenge in protein secondary structure prediction research, as also highlighted by Zhou and Troyanskaya (2014) and Ismi et al. (2022). The resampling techniques are not employed in this research due to resource and time limitation. Incorporating class balancing techniques and utilizing transfer learning from larger protein datasets could further improve model performance.

This research demonstrates the potential of NLP techniques in protein secondary structure prediction but highlights the need for further improvement. A significant limitation of protein language models reviewed in the literature survey, is the high computational resource and time. Even using low resource techniques, this research still faced resource exhaustion during the embedding extraction. Significant improvements can be done by including other protein features

like PSSM and HMM along with the embeddings to improve the accuracy across all classes as depicted in papers reviewed in the literature section, Chen et al. (2016) and Sofi and Wani (2022).A similar study by Jin et al. (2021) successfully combined ProtTrans embeddings with PSSM and HHM profiles, achieving notable model improvements. Another significant improvement that can be done is the enabling of ensemble model with combination of different deep learning model and different protein input along with these NLP embeddings can improve the prediction as highlighted by Singh et al. (2022) and Wang et al. (2019), which is also clearly stated in section 2. The insights on combining static embeddings with protein-specific features and ensemble models encourage academic researchers to experiment with hybrid methodologies for improved accuracy and robustness.
This research demonstrates the feasibility of using computationally less expensive NLP techniques, such as Word2Vec, for protein secondary structure prediction. Additionally, the research identifies the limitations of static embeddings in extracting protein features while enhancing the use of attention-based mechanisms along with ESM for improving the predictions. However, the research shows the persistent challenge of class imbalance in protein datasets, contributing to the necessity of incorporating class balancing techniques. The research provides potential impact on both medical practitioners and academia. The computationally less expensive NLP techniques makes it possible for medical and research practitioners to perform protein structure predictions faster and cost effectively. Also, it improves the accessibility in resource-constrained settings like small labs or hospitals. Moreover, the study contributes to the feasibility of NLP techniques in bio-informatics, for exploring lightweight approaches in protein related tasks. This research enables interdisciplinary collaboration, allowing computer scientists, bioinformaticians, and molecular biologists to work together in advancing protein prediction techniques by adapting NLP models for protein sequences.

7 Conclusion and Future Work

The research focuses on investigating how is the prediction of eight-state protein secondary structure influenced by deep learning models combined with NLP techniques for sequence analysis and feature extraction. The NLP techniques such as - Word2vec, Glove and ESM are used with LSTM model for investigating the effect. The Word2Vec with LSTM, GloVe with LSTM, and ESM with LSTM models achieved overall accuracies of 91.89%, 83.62%, and 56.14%, respectively. However, all three models only identified the classes B, E, H, and L, highlighting that embeddings generated from these techniques alone are insufficient for predicting all eight classes. Notably, the ESM with LSTM model demonstrated improved performance on previously unidentified classes when combined with a BiLSTM and attention mechanism.

In future, the research can be extended by exploring the embeddings from these models along with other protein features like PSSM. Moreover, resampling or class-balancing techniques, such as SMOTE or cost-sensitive learning, can address class imbalance challenges effectively. Additionally, experimenting the embeddings with hybrid architectures such as ensemble methods or transformers, could improve the prediction of underrepresented classes and overall model performance.

References

Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations (ICLR)*. ICLR, 2019. URL https://arxiv.org/abs/1902.08661.

Yehong Chen, Jinyong Cheng, Yihui Liu, and Pil Seong Park. A novel approach of protein secondary structure prediction by svm using pssm combined by sequence features. In *Proceedings of the SAI Intelligent Systems Conference (IntelliSys 2016)*, pages 1074–1084. Springer International Publishing, 2016. doi: 10.1007/978-3-319-56994-9_74.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yuan Wang, Lynn Jones, Thomas Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high-performance computing. *arXiv preprint arXiv:2007.06225*, 2020. doi: 10.48550/ARXIV.2007.06225.

Soumen Ghosh and Pintu Chandra Shill. Protein secondary structure detection without alignment by recurrent neural network with lstm. In *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6, 2021. doi: 10.1109/EICT54103.2021.9733530.

Michael Heinzinger, Ahmed Elnaggar, Yuan Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(1):723, 2019. doi: 10.1186/s12859-019-3220-8.

Magnus Haraldson Høie, Erik Nicolas Kiehl, Bent Petersen, Morten Nielsen, Ole Winther, Henrik Nielsen, Jeppe Hallgren, and Paolo Marcatili. Netsurfp-3.0: accurate and fast prediction of protein structural features by protein language models and deep learning. *Nucleic Acids Research*, 50(W1):W510–W515, 06 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac439. URL https://doi.org/10.1093/nar/gkac439.

Dewi Pramudi Ismi, Reza Pulungan, and Afiahayati. Deep learning for protein secondary structure prediction: Pre and post-alphafold. *Computational and structural biotechnology journal*, 20:6271—6286, 2022. ISSN 2001-0370. doi: 10.1016/j.csbj.2022.11.012. URL https://europepmc.org/articles/PMC9678802.

Kanchan Jha, Sriparna Saha, and Sourav Karmakar. Prediction of protein-protein interactions using vision transformer and language model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(5):3215–3225, 2023. doi: 10.1109/TCBB.2023.3248797.

Hailong Jin, Wei Du, Jiawei Gu, Tianhao Zhang, and Xiaohu Shi. Combining gcn and bi-lstm for protein secondary structure prediction. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 44–49, 2021. doi: 10.1109/BIBM52615.2021.9669366.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983. doi: 10.1002/bip.360221211.

Zeming Lin, Jack Lanchantin, and Yanjun Qi. Must-cnn: A multilayer shift-and-stitch deep convolutional architecture for sequence-based protein structure prediction, 2016. URL https://arxiv.org/abs/1605.03004.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

M. M. Mohamed Mufassirin, M. A. Hakim Newton, Julia Rahman, and Abdul Sattar. Multi-s3p: Protein secondary structure prediction with specialized multi-network and self-attention-based deep learning model. *IEEE Access*, 11:57083–57096, 2023. doi: 10.1109/ACCESS.2023.3282702.

Dan Ofer, Nadav Brandes, and Michal Linial. The language of proteins: Nlp, machine learning protein sequences. *Computational and Structural Biotechnology Journal*, 19: 1750–1758, 2021. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2021.03.022. URL `https://www.sciencedirect.com/science/article/pii/S2001037021000945`.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1162. URL `https://www.aclweb.org/anthology/D14-1162`.

Arifur Rahman, Anik Mahmud, and Pintu Chandra Shill. Neural network-based approach to predict protein secondary structure. In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 135–140, 2023. doi: 10.1109/ICAAIC56838.2023.10140404.

Jaspreet Singh, Kuldip Paliwal, Thomas Litfin, Jaswinder Singh, and Yaoqi Zhou. Reaching alignment-profile-based accuracy in predicting protein secondary and tertiary structural properties without alignment. *Scientific Reports*, 12(1):7607, 2022. doi: 10.1038/s41598-022-11684-w. URL `https://doi.org/10.1038/s41598-022-11684-w`.

Soubraylu Sivakumar, Lakshmi Sarvani Videla, T Rajesh Kumar, J. Nagaraj, Shilpa Itnal, and D. Haritha. Review on word2vec word embedding neural net. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pages 282–290, 2020. doi: 10.1109/ICOSEC49089.2020.9215319.

Mukhtar Ahmad Sofi and M. Arif Wani. Improving prediction of protein secondary structures using attention-enhanced deep neural networks. In *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 664–668, 2022. doi: 10.23919/INDIACom54597.2022.9763114.

CS Srushti, PM Prathibhavani, and KR Venugopal. Eight-state accuracy prediction of protein secondary structure using ensembled model. In *2023 International Conference for Advancement in Technology (ICONAT)*, pages 1–6. IEEE, 2023.

Jian Wang, Jinyong Cheng, Zhigang Zhao, and Wenpeng Lu. Protein secondary structure prediction using ensemble of lstm neural networks. In *2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 2019. doi: 10.1109/ICISCAE48440.2019.221626.

Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, 6:18962, 2016. doi: 10.1038/srep18962.

Wen-Wu Zeng, Ning-Xin Jia, and Jun Hu. Improved protein secondary structure prediction using bidirectional long short-term memory neural network and bootstrap aggregating. In *2022 10th International Conference on Bioinformatics and Computational Biology (ICBCB)*, pages 33–40, 2022. doi: 10.1109/ICBCB55259.2022.9802482.

Jian Zhou and Olga G. Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32, Beijing, China, 2014. JMLR: W&CP. URL `https://proceedings.mlr.press/v32/zhou14.pdf`.