APPLICATION OF LSTM FOR PROTEIN PROTEIN INTERACTION PREDICTION

Project Submitted to the
SRM University AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology
in
Computer Science & Engineering
School of Engineering & Sciences

submitted by

Manogna Grandhi(AP20110010627)

Roshitha Tiruveedhula(AP20110010586)

Sreeshanth Edam(AP20110010589)
Satwik korrapati(AP20110010592)

Under the Guidance of

Dr.Anirban Bhar

Department of Computer Science & Engineering
SRM University-AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240
May 2024

DECLARATION

I undersigned hereby declare that the project report **Application of LSTM for Protein protein interaction prediction** submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology in the Dr.Anirban Bhar, SRM University-AP, is a bonafide work done by me under supervision of Prof. Anirban Bhar. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree of any other University.

Place	:	....	Date	:	May 16, 2024
Name of student	:	Manogna Grandhi	Signature	:	....
Name of student	:	Roshitha Tiruveedhula	Signature	:	....
Name of student	:	Sreeshanth Edam	Signature	:	....
Name of student	:	Satwik korrapati	Signature	:	....

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRM University-AP

Neerukonda, Mangalgiri, Guntur

Andhra Pradesh - 522 240

CERTIFICATE

This is to certify that the report entitled **Application of LSTM for Protein protein interaction prediction** submitted by **Manogna Grandhi, Roshitha Tiruveedhula, Sreeshanth Edam, Satwik korrapati** to the SRM University-AP in partial fulfillment of the requirements for the award of the Degree of Master of Technology in in is a bonafide record of the project work carried out under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Guide	Head of Department
Name : Dr.Anirban Bhar	Name : Prof. Niraj Upadhyaya
Signature:	Signature: ....

ACKNOWLEDGMENT

We would like to sincerely thank everyone that helped us with the development and delivery of this project report, which is entitled **Application of LSTM for Protein protein interaction prediction**

We would especially like to thank Dr.Anirban Bhar, our supervisor and guide from the Department of Computer Science & Engineering for all of their helpful advice and important contributions, which significantly improved the quality of our report. We also appreciate the ongoing support from Prof. Niraj Upadhyaya, the Head of Department of Computer Science & Engineering

We owe our friends and classmates a debt of gratitude for their steadfast support during this undertaking. Their willingness to listen to our presentations and their helpful criticism have been much appreciated.

Manogna Grandhi, Roshitha Tiruveedhula, Sreeshanth Edam, Satwik korrapati
(Reg. No. AP20110010627, AP20110010586, AP20110010589, AP20110010592)
B. Tech.
Department of Computer Science & Engineering
SRM University-AP

ABSTRACT

Numerous protein–protein interactions (PPIs) occur between the person and the virus during viral infection. PPIs can cause host transcription machinery to be hijacked or for viral coat proteins to first connect to host membrane receptors. Unfortunately, because to the time-consuming and costly nature of experimental techniques like mass spectrometry and the restriction of molecular dynamic modeling to proteins with solved 3D structures, only few interspecies PPIs have been found. It is anticipated that sequence-based machine learning techniques would solve these issues. Using only amino acid sequences, we have created the LSTM-PHV model, which uses word2vec to predict PPIs between humans and viruses. With a very unbalanced ratio of positive to negative samples, the LSTM-PHV successfully learned the training set. On the independent and training datasets, it obtained accuracies of 0.830141 and 0.694 and AUCs of 0.903425 and 0.74436, respectively. Compared to previous state-of-the-art PPI predictors, the LSTM-PHV learnt significantly excelled at predicting PPIs between human and unknown or novel viruses. It's interesting to note that PPI prediction may be achieved by just memorizing word sequence contexts. The LSTM-PHV was able to identify the positive PPI samples from the negative ones with great clarity because to the use of uniform manifold approximation and projection.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Organic molecules called proteins are found in biological systems and perform a variety of special tasks such storage, transportation, and membrane composition[1]. They also reveal physiological and biological activities by interacting with rna, dna, and other proteins in a similar way[2]. Different types of circumstances are required for this interaction. The first is design-based, with a specified biomolecular event as the output, and the second is all about developing the non-generic function. Thus, the prediction of protein three-dimensional structures has benefited greatly from research on protein-protein interactions and protein ligand blinding issues. Additionally, protein-protein interactions aid in the complexing of several biological processes and activities, including metabolic cycles, signal transmission, and dna transcriptions. Chronic degenerative illnesses and cancer are caused by interactions between proteins in cells and their activities[3].It is also useful for creating antimicrobial medications. While experimenting, conventional biological methods like affinity purifications and two hybrid screenings can be used to identify protein interaction sites; however, these processes are costly and time-consuming, which makes it helpful to predict accurate calculation methods to predict protein interaction sites. The purpose of lstm is to construct a deep neural network that can predict whether or not a DNA sequence may bind to a protein. It does this by solving the DNA binding process, which has a specific affinity for either single- or double-stranded DNA. There are several challenges in predicting protein-protein interactions; the simplest method to get over these is to use lstm, which predicts values with accuracy[4]. This LST component uses a weighted loss function to lessen the interference caused by unbalanced data, and it can capture both spatial and sequential characteristics. In conclusion, this research discusses how to predict correct values using LSTM and predict protein-protein interactions.

Chapter 2

MOTIVATION

2.1 IMPORTANCE OF PROTEIN PROTEIN INTERACTION(PPI) PREDICTION

Proteins are vital components of biological systems because they carry out a variety of tasks include transporting molecules, initiating biochemical events, and supporting structural integrity. Protein-protein interactions (PPIs), which are interactions between proteins, are responsible for a large number of these processes[5]. Comprehending PPIs plays a crucial role in deciphering the intricate web of biological activities occurring within cells and clarifying the causes behind diverse ailments. Co-immunoprecipitation and yeast two-hybrid experiments are two examples of labor-intensive, time-consuming, and costly experimental techniques for identifying PPIs. Furthermore, it's possible that these techniques don't fully represent the intricacy of PPI networks in real cells. As a result, there is increasing interest in creating computational techniques that use protein sequence data to predict PPIs. Deep learning is ideally suited for PPI prediction since, in particular, LSTM networks have shown to be an effective tool for sequential data analysis. Long-range relationships in protein sequences may be captured by LSTM networks, which is useful for comprehending the structure and function of proteins[6]. The purpose of this research is to provide a more effective and economical approach to PPI prediction by utilizing the capabilities of LSTM networks. If LSTM networks can accurately predict PPIs, it might have a significant influence. It could result in the identification of novel protein interactions, offering fresh perspectives on the pathophysiology of illnesses including cancer, neurological conditions, and infectious diseases. Furthermore, it could make it easier to create tailored medicines that interfere with certain PPIs implicated in disease processes, resulting in more efficient and individualized care. All things considered, the aim of this study is to boost our knowledge of intricate biological systems by creating computational techniques for PPI prediction. Our goal is to help identify novel therapeutic targets and provide cutting-edge remedies for a variety of illnesses by fusing the strengths of bioinformatics and deep learning.

2.2 PROBLEM STATEMENT

The majority of biological activities, such as signal transmission, metabolic pathways, and immunological responses, depend on protein-protein interactions, or PPIs. Progressing our understanding of biological activities and disease pathways requires an understanding of these interactions. Nevertheless, PPIs are frequently determined by expensive, time-consuming, and difficult experimental approaches. For forecasting PPIs, computational method especially those based on machine learning offer a viable substitute.

2.2.1 Proposed solution.

The suggested remedy is creating a real-time prediction system that predicts PPIs from protein sequence data by using LSTM. Recurrent neural networks (RNNs) of the long short-term memory (LSTM) variety are excellent at recognizing relationships in sequential data, which makes them a good fit for examining protein sequences, which are just chains of amino acids.

Protein sequence data would be continually gathered by the system from public databases or experimental sources, and then fed into the LSTM network for prediction. In order to identify the intricate patterns and dependencies in protein sequences that are suggestive of interactions, the LSTM network would be trained on a dataset of known PPIs.

2.2.2 Key Features

1. Real-Time Prediction: For scientists and medical professionals, the system's capacity to offer real-time predictions of protein-protein interactions (PPIs) is essential. In scientific and clinical contexts, real-time predictions facilitate prompt analysis and interpretation of experimental outcomes, hence facilitating fast decision-making. Real-time predictions, for instance, might be used by researchers examining medication interactions to pinpoint possible targets for new drug development.

2. Continuous Learning: Increasing prediction accuracy over time depends on the system's capacity to update its predictions on a regular basis in light of new information. The system may improve its forecasting power by adding additional protein sequences and interaction data to its model as they become available. By keeping the system updated with the most recent study findings, continuous learning makes sure that it keeps becoming better over time.

3. Integration with Experimental Data: By integrating the system with experimental data, researchers may improve the model and confirm predictions. Researchers may evaluate the prediction accuracy and pinpoint areas for improvement by contrasting the system's predictions with experimen-

tal data. Integrating with experimental data also helps researchers confirm ideas generated by the system and obtain a greater grasp of the underlying mechanics of protein interactions.

4. User-Friendly Interface: Researchers may enter protein sequences, see predictions, and analyze results with ease thanks to the system's user-friendly interface. To meet the varied demands of users, the interface may have elements like adjustable settings, interactive visualizations, and data filtering choices. The system's usability and accessibility are improved by an intuitive interface, increasing its broad use in clinical and research contexts.

All in all, these essential components support the system's real-time protein-protein interaction prediction efficacy and ability to propel biological and medical research forward. The system facilitates the rapid development of novel therapies and treatments by allowing researchers and clinicians to gain valuable insights into protein interactions through real-time predictions, continuous learning from new data, integration with experimental data, and an intuitive interface.

2.2.3 Benefits

1. Research Acceleration: Predicting protein-protein interactions (PPIs) in real time will greatly speed up biological and medical research. Instantaneous insights into protein interactions enable researchers to swiftly formulate theories, plan tests, and confirm findings. Research is moving more quickly than ever, which may help us grasp intricate biological processes and create novel treatments.

2. Increased Accuracy: PPI predictions may be made more precisely by using LSTM, which are able to recognize relationships in protein sequences. This increased precision is essential for expanding our knowledge

of protein interactions and can reveal new interactions that would not have been found with just conventional experimental techniques.

3. Cost-effective: By accurately predicting PPIs, the real-time prediction system may eliminate the need for expensive experimental techniques. Conventional experimental techniques for identifying PPIs can be expensive, time-consuming, and labor-intensive. Examples include yeast two-hybrid tests and co-immunoprecipitation. Researchers can save money by prioritizing experiments and concentrating resources on confirming expected interactions when they depend on computer predictions.

4. Possibility for Drug research: Precise PPI prediction has the potential to revolutionize drug research by identifying new pharmacological targets and illness-treating regimens. Numerous diseases, including cancer, neurological disorders, and infectious infections, are primarily brought on by dysregulated protein interactions. Accurately predicting PPIs allows researchers to identify potential targets for developing drugs and produce more potent therapies.

Conclusively, the use of LSTM networks for real-time PPI prediction has several noteworthy advantages, such as expedited research, enhanced precision, economical viability, and the possibility of novel medication discoveries. By taking use of these advantages, scientists and medical professionals may learn important lessons about protein interactions, expand our comprehension of intricate biological processes, and create novel treatments for a variety of illnesses.

Chapter 3

LITERATURE SURVEY

As the processing power has increased over the past 20 years, machine learning has being used more and more. As a result, it is only recently being used to predict biological sequence. Ahmed et al.[7] collected traits of triplets and quadruplets to train different models, such as neural networks and support vector machines. The most favorable one was chosen for forecasting. The neural network was made up of four layers, each of which included a varied number of input and hidden level nodes. With this configuration and 60,000 training and testing data points, a region under the curve analysis of 0.92 was obtained. Tsukiyama et al.[8] developed the LSTM model, or LSTM PHV, utilizing word2vec to predict PPIs between humans and viruses using just amino acid sequences, which is comparable to our model. The LSTM-PHV obtained an accuracy of 98.4 percentage and an AUC of 0.976 by effectively learning the training data with a highly skewed ratio of positive to negative samples through the use of 5-fold cross-validation. The LSTM-PHV outperformed existing predictors in predicting PPIs between humans and unknown or new viruses after being trained on several host protein-inclusive datasets.

It's interesting to note that the performance was exceptionally good when the "words" were limited to amino acid sequence contexts. The LSTM-PHV significantly separated the positive PPI data from the negative ones, as shown by the use of uniform manifold approximation and projection. This model's structure is comparable to the "4D Bi-LSTM doubleip" model that we describe in our work. In order to facilitate PPI extraction, Zhou et al.[9] employed knowledge bases (KBs), which include enormous quantities of structured data on protein entities and their relationships. These KBs may be expressed in entity and relation embeddings. To make the preceding understanding of protein–protein pairings appropriate in various situations, it must be applied carefully. Another research, called PPIMem(Öztürk et al., 2018), used primary sequence data and protein evolutionary histories with LSTM networks to predict PPIs. PPIMem improved prediction accuracy by capturing long-range relationships in protein sequences by taking evolutionary information into account. This method shows how useful it is to use evolutionary data in PPI prediction models.

To predict PPIs, a new method called DeepPPI[10] used LSTM and Convolutional Neural Networks (CNNs). DeepPPI's prediction performance was enhanced by capturing both local and global sequence trends through the integration of structural characteristics and protein sequence information. The significance of merging several neural network architectures for more accurate PPI prediction is highlighted by this fusion of CNN and LSTM.

Moreover, DeepPPI-FNN(Chen et al., 2019)[11] predicted PPIs using a fusion neural network (FNN) architecture that contains LSTM layers. DeepPPI-FNN increased its prediction ability by fusing domain-domain interactions with knowledge about protein sequences. This method emphasizes how crucial it is to combine data from several sources in order to predict PPI more precisely.

In conclusion, by combining different kinds of data and characteristics, LSTM networks have been effectively used to forecast PPIs. These findings show how LSTM networks may effectively capture intricate interactions seen in interaction networks and protein sequences, improving prediction accuracy. LSTM networks will probably be crucial in expanding our knowledge of PPIs and their function in biological processes as bioinformatics develops.

Chapter 4

METHODOLOGY

4.1 DATASET

There are 3 csv files named dataset_independent_test, positive and negative samples where each dataset contains the rows and columns mentioned below and the dataset independent test dataset contains one more column named labels and it consists of values either 0 or 1 representing 1 as a protein protein interaction and 0 representing no protein protein interaction between them.

Filename	***Content***
Human_pro	Transformed vector produced during the extraction of features from human proteins by the LSTM-PHV network. The transformed vector in a sample is contained in each row.
Virus_pro	while extracting features from the viral protein, the LSTM-PHV network generated a transformed vector. The transformed vector in a sample is contained in each row
Human_seq	weights for attention produced by the LSTM-PHV network during the extraction of features from human proteins. The attention weights in a sample are listed in each row.
Virus_seq	weights for attention produced when the LSTM-PHV network extracts features from viral proteins. The attention weights in a sample are listed in each row

Table 4.1: Dataset

4.2 USING WORD2VEC EMBEDDING MODEL TO ENCODE PROTEIN SEQUENCES

Word vectors are produced by Word2Vec and are distributed numerical representations of word attributes. These word qualities might include terms that convey the significance of the context around each individual word that appears in our lexicon. Word embeddings ultimately aid in identifying a word's link to another word with a comparable meaning through the created vectors. Word2Vec is a common natural language processing strategy based on the idea that words have similar meanings when they appear in comparable contexts [12]. Distributed vector representations of words are the outcome of this. Continuous Bag-of-Words (CBOW) and Skip-Gram are its two main types.Both models are trained on large text corpora in order to acquire word embeddings, which are dense, high-dimensional vectors that capture the semantic relationships between words. These embeddings may be applied to do deductive reasoning, measure word similarity, and improve performance on various NLP tasks. In this work, amino acid sequences—the building blocks of proteins—were analyzed and represented using the concepts of Word2Vec, a technique mainly employed in natural language processing (NLP)[12]. Long sequences of amino acids make up proteins, and knowing the interactions between these amino acids is essential to understanding how proteins operate. The idea of k-mers was taken by the researchers to apply Word2Vec to amino acid sequences. K-mers are k-length subsequences that are part of a longer sequence. The amino acid sequences were split into k-mers in this case, where a k-mer is a distinct length of k consecutive amino acids.

For example, the amino acid sequence ”GATCGATCGA” would be split into four 4-mers if k = 4. These would be ”GATC,” ”ATCG,” ”TCGA,”

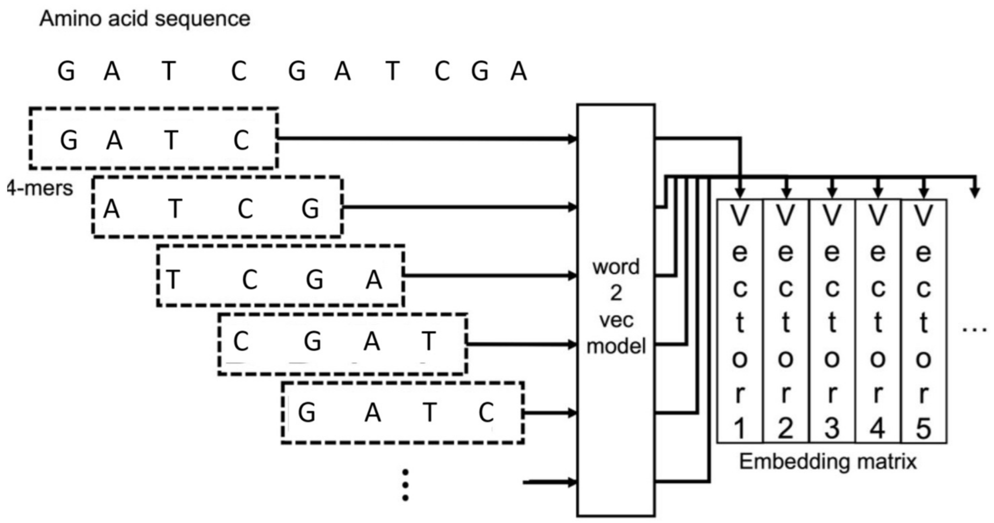


Figure 4.1: Example of 4-mer Embedded amino acid sequences.

and "CGAT." Similar to how words are handled in NLP, each of these k-mers is considered as a "word" in the Word2Vec architecture. The researchers were able to utilize Word2Vec to learn and record the contextual links between these k-mers, which could subsequently be used to comprehend the structural and functional features of proteins, by modeling amino acid sequences as sequences of k-mers. This method gives researchers a fresh viewpoint on researching protein sequences and interactions by utilizing NLP techniques to get insights into the complicated language of proteins.

4.3 CONSTRUCTION OF LSTM-PHV

Strong neural networks, such convolutional neural networks (CNN) and recurrent neural networks (RNN), have been used to tackle challenging issues like voice recognition and visual object identification. Training on variable-length data is made possible by the RNN, which also learns time or step relationships in sequence data. By resolving the gradient explosion and disappearance issues with RNNs, the LSTM makes long-term time-dependent learning possible.

The figure 4.2 is the structure of LSTM-PHV,The human and viral protein matrices were transformed into the two fixed-length vectors, respectively, using the two upstream neural networks employing the LSTM. Lines of blue and red encircled the networks. . At every stage, the LSTM unit received feature vectors for every 4-mer in the protein matrices. At each phase, the output from the LSTM unit was subjected to three completely linked layers in order to produce scalar values. The output from the LSTM unit and the broadcasted scalar values were multiplied. The vectors in human and virus were multiplied and concatenated to create the fixed-length vectors. Four fully linked layers produced the final outputs that are shown by the purple line.

A thorough and methodical approach to model construction and training is demonstrated by the inclusion of many crucial processes and parameters in the training of the LSTM-PHV model for protein-protein interaction (PPI) prediction.

For efficient PPI prediction, the LSTM architecture and the Protein Histone Vector (PHV) encoding technique are combined in the LSTM-PHV model. Protein amino acid sequences are encoded into fixed-length vector representations using the PHV approach, which helps the model recognize significant patterns and characteristics in the sequences. A dataset comprising protein sequences, interaction labels, and human and viral protein IDs is utilized to train the LSTM-PHV model. To assess the performance of the model, the dataset is split into training and validation sets. Each row in the training data represents a protein pair and its associated label (1 for interact,

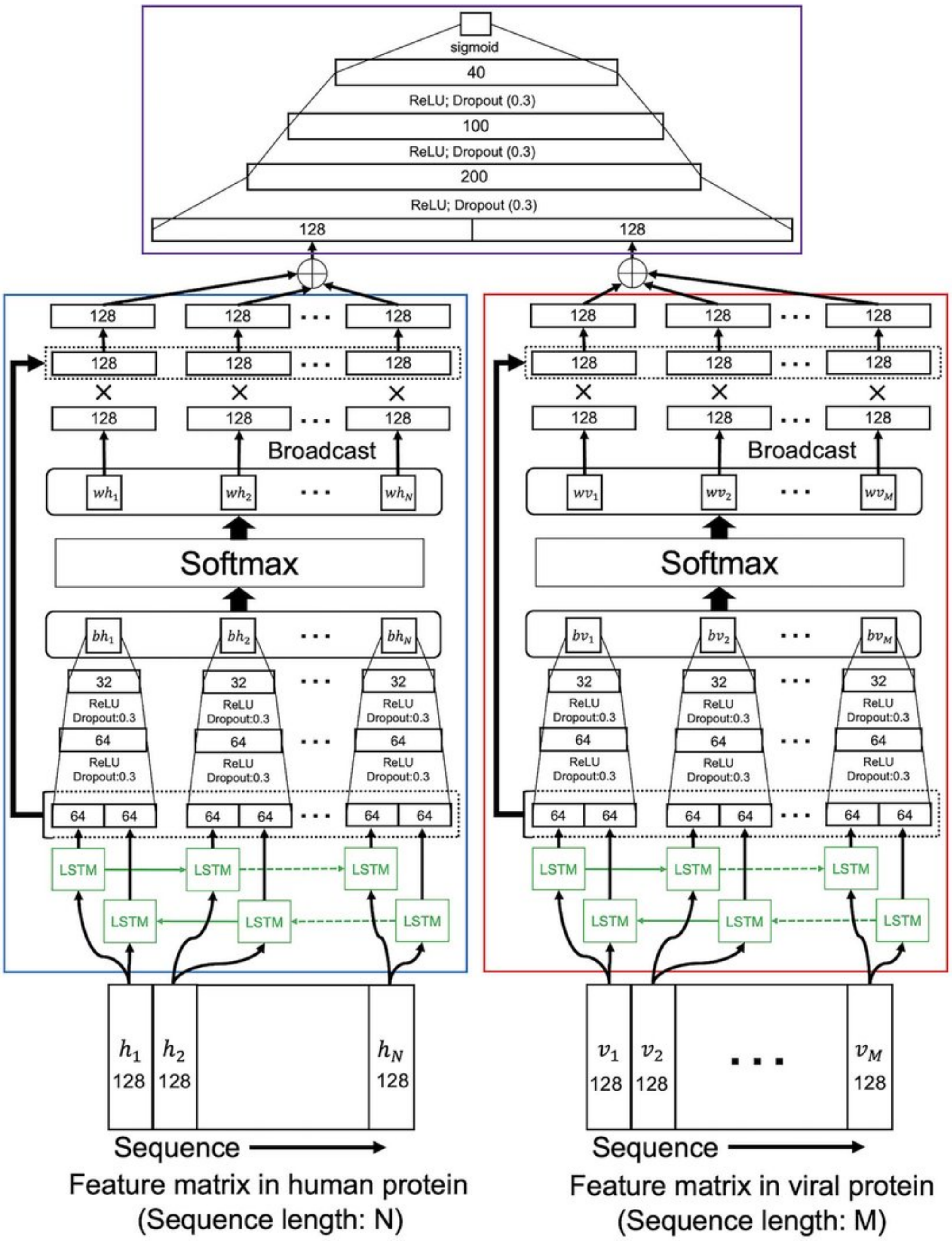


Figure 4.2: Structure of LSTM-PHV [8]

0 for not interact). The data is supplied in CSV format.

Using a pre-trained Word2Vec model to convert the protein sequences into vector representations is a crucial step in the training process. In order to anticipate protein interactions, it is imperative that the semantic links between the amino acids in the sequences be captured. LSTM layers are used in a deep learning method to train the model. Since LSTM layers are good at capturing long-term relationships, they are a good fit for sequential data, such as protein sequences. A predetermined learning rate (usually set to a low value like 0.001) is used by the model to optimize a loss function using an optimization algorithm during training in order to reduce the discrepancy between the predicted interaction scores and the real labels.

There are two possibilities for the loss function—"imbalanced" and "balanced"—to handle imbalanced data. When there is a notable class imbalance in the dataset—that is, when the proportion of positive (interacting) samples to negative (non-interacting) samples is much lower—the "imbalanced" loss function is employed. When the dataset is more balanced, with nearly equal numbers of negative samples and positive samples, the "balanced" loss function is employed.

Early stopping is used to prevent overfitting during the training phase, which iterates over a predetermined number of epochs. Early stopping terminates the training process if, after a certain number of epochs, the model's performance on the validation set does not improve. The trained LSTM-PHV model and the training process are stored to the specified output directory after training. The trained LSTM-PHV model is stored in the "deep_model" model file, and the learning process, including accuracy and loss metrics throughout epochs, is documented in the "deep_HV_result.txt". These files can be utilized for additional research and assessment as they offer insightful information about the model's performance. Overall, the LSTM-PHV model's training for PPI prediction demonstrates a thorough approach to model building and training, emphasizing the value of utilizing cutting-edge methods in protein sequence analysis and deep learning.

4.4 EVALUATION METRICS

Seven statistical metrics were employed to assess the prediction performance: F1-score, AUC, accuracy, precision, sensitivity (recall), specificity, accuracy, Matthews correlation coefficient (MCC), precision, and area under the precision-recall curve (AUPRC). AUPRC, F1-score, and MCC are useful metrics for evaluating unbalanced data. The metrics provided, apart from the AUC and AUPRC, are:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TN} + \text{FN}) \times (\text{TP} + \text{FP}) \times (\text{TN} + \text{FP}) \times (\text{TP} + \text{FN})}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F1} = \frac{2\times \text{ precision } \times \text{ recall}}{\text{precision } + \text{ recall}}$$

where the numbers for correctly anticipated positive samples, incorrectly estimated positive samples, correctly projected negative samples, and incorrectly predicted negative samples are represented by the symbols TP, FP, TN, and FN, respectively. Protein pair interaction thresholds were established at a predicted probability of 0.5. The areas beneath the ROC and PR curves are called AUC and AUPRC, respectively. The Python program Scikit-learn was utilized to compute these measurements.

Chapter 5

IMPLEMENTATION

5.1 LSTM-PHV FEATURES

1. Amino Acid Sequence Prediction: Based only on amino acid sequences, LSTM-PHV predicts PPIs. When compared to other methods that might need additional structural or biochemical data, this allows for a simpler input format.

2. Weights of Attention: Attention weights are included in the package, which is useful for locating protein binding interfaces. The attention mechanisms in the model allow it to concentrate on pertinent segments of the input sequence, which may improve prediction accuracy by highlighting important amino acids that are involved in interactions.

5.2 SETUP OF ENVIRONMENT

It is advised to set up a specific environment in order to manage dependencies efficiently when using LSTM-PHV.

For this purpose, a well-known Python distribution called Anaconda is recommended. This is a quick rundown of how the environment is set up:

1. Building Virtual Environments: Users can establish a virtual environment specifically for LSTM-PHV using Anaconda. In order to avoid conflicts with other installed packages, this isolates the package and its dependencies from the Python environment across the entire system.

2. Start-up: After the virtual environment has been created, the conda activate command must be used to activate it. This guarantees that the executions and subsequent installations take place in the isolated environment.

5.3 STEPS IN PROCESSING

Three main processing tasks are supported by the Command-Line Interface (CLI) system offered by LSTM-PHV:

1. Word2Vec Embedding Model Training: A Word2Vec embedding model is trained first, and then the LSTM-PHV model is trained for PPI prediction. This model captures semantic similarities between amino acids by encoding amino acid sequences into distributed representations.

2. LSTM-PHV Model Training: The encoded amino acid sequences and corresponding labels are used to train the primary LSTM-PHV model for PPI prediction. In order to train the LSTM network to recognise patterns and features suggestive of protein interactions, this step entails using attention mechanisms.

3. PPI Forecast: The LSTM-PHV model can be used to predict PPIs for novel input sequences after it has been trained. Based on the amino acid sequences of the proteins, the model assesses the interactions between them, providing

5.4 SETTING UP AND INSTALLING

Users must take the following actions in order to install the package and set up the environment for LSTM-PHV:

1. Setting Up the Virtual Environment: Optionally, use Anaconda to build a virtual environment where LSTM-PHV and its dependencies are

isolated.

2. Installing the package: Use pip to install the LSTM-PHV package by giving it the path to the package distribution file. This sets up the package in the virtual environment and installs all of its dependencies.

3. Word2Vec Model Training: Following installation, users can specify the output directory for the trained model and supply the training data in FASTA format to train the Word2Vec embedding model.

By utilising deep learning methods like LSTM networks and attention mechanisms, LSTM-PHV provides an all-encompassing approach to PPI prediction from amino acid sequences. By adhering to the guidelines given, users

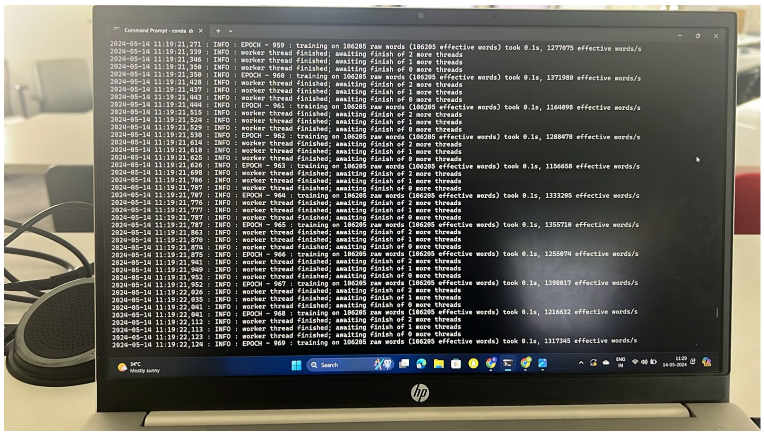
can set up the environment, train models, and make predictions effectively for protein-protein interaction studies.

Chapter 6

HARDWARE/ SOFTWARE TOOLS USED

6.1 HARDWARE TOOLS USED

64-bit operating system, x64-based processor was used for compiling the output of the project.



```
2024-05-14 11:19:21,271 : INFO : EPOCH - 959 : training on 106205 raw words (106205 effective words) took 0.1s, 1277075 effective words/s
2024-05-14 11:19:21,339 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,346 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,350 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,350 : INFO : EPOCH - 960 : training on 106205 raw words (106205 effective words) took 0.1s, 1371980 effective words/s
2024-05-14 11:19:21,428 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,437 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,443 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,444 : INFO : EPOCH - 961 : training on 106205 raw words (106205 effective words) took 0.1s, 1164098 effective words/s
2024-05-14 11:19:21,515 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,524 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,529 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,530 : INFO : EPOCH - 962 : training on 106205 raw words (106205 effective words) took 0.1s, 1288478 effective words/s
2024-05-14 11:19:21,614 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,618 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,625 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,626 : INFO : EPOCH - 963 : training on 106205 raw words (106205 effective words) took 0.1s, 1156658 effective words/s
2024-05-14 11:19:21,698 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,706 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,707 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,707 : INFO : EPOCH - 964 : training on 106205 raw words (106205 effective words) took 0.1s, 1333205 effective words/s
2024-05-14 11:19:21,776 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,777 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,787 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,787 : INFO : EPOCH - 965 : training on 106205 raw words (106205 effective words) took 0.1s, 1355710 effective words/s
2024-05-14 11:19:21,863 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,870 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,874 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,875 : INFO : EPOCH - 966 : training on 106205 raw words (106205 effective words) took 0.1s, 1255074 effective words/s
2024-05-14 11:19:21,941 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:21,949 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:21,952 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:21,952 : INFO : EPOCH - 967 : training on 106205 raw words (106205 effective words) took 0.1s, 1398017 effective words/s
2024-05-14 11:19:22,026 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:22,035 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:22,041 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:22,041 : INFO : EPOCH - 968 : training on 106205 raw words (106205 effective words) took 0.1s, 1216632 effective words/s
2024-05-14 11:19:22,112 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 11:19:22,113 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 11:19:22,123 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 11:19:22,124 : INFO : EPOCH - 969 : training on 106205 raw words (106205 effective words) took 0.1s, 1317345 effective words/s
```

Figure 6.1: Caption

6.2 SOFTWARE TOOLS USED

We have used anaconda command prompt which is a command-line interface for the Anaconda distribution.It provides us a way to manage packages, environments, and execute efficiently.With its specialized tools, it streamlines tasks like package installation, environment creation, and version control. We also used Torchvision which is a computer vision library in PyTorch, offering tools for image and video processing, including datasets, transforms, and models. Software's and their version:

biopython==1.78

numpy==1.19.2

gensim==3.8.3

torch==1.7.1

torchvision==0.8.2

torchaudio==0.7.2

torch_optimizer==0.1.0

scikit-learn==0.24.1

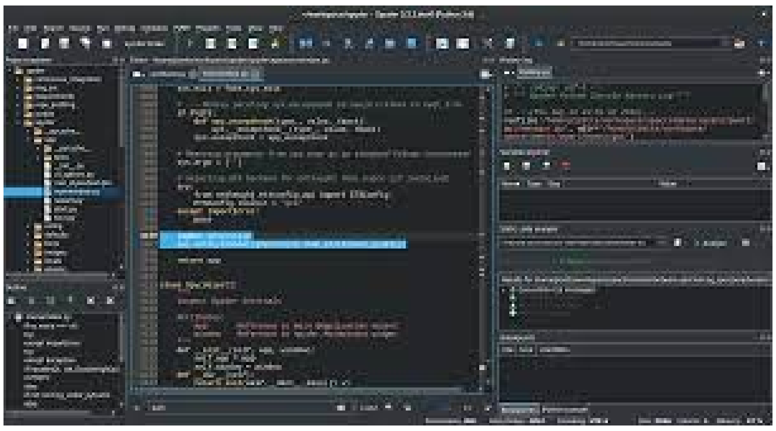


Figure 6.2: Python IDE

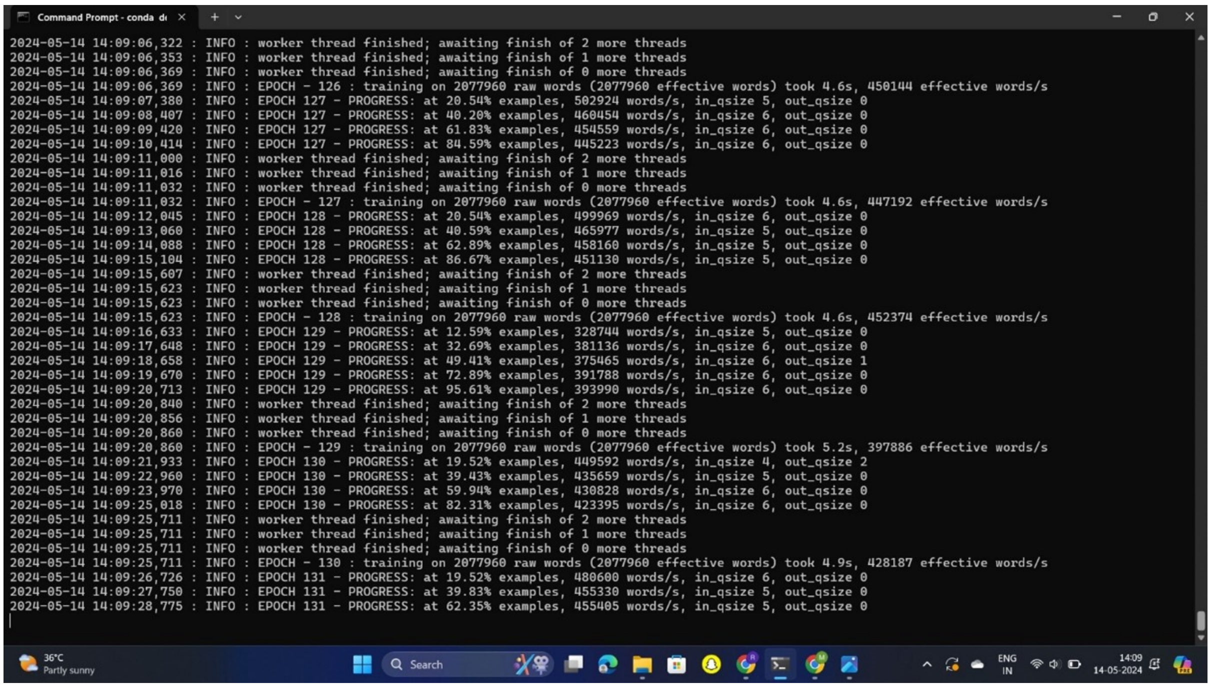
Chapter 7

RESULTS & DISCUSSION

Our study on Protein-Protein Interactions (PPIs) prediction using Long Short-Term Memory (LSTM) networks is presented in the results section. By comparing the predictions of the LSTM-PHV model with known protein-protein interactions, we assess the model's performance using measures- F1-score, accuracy, recall, and precision.

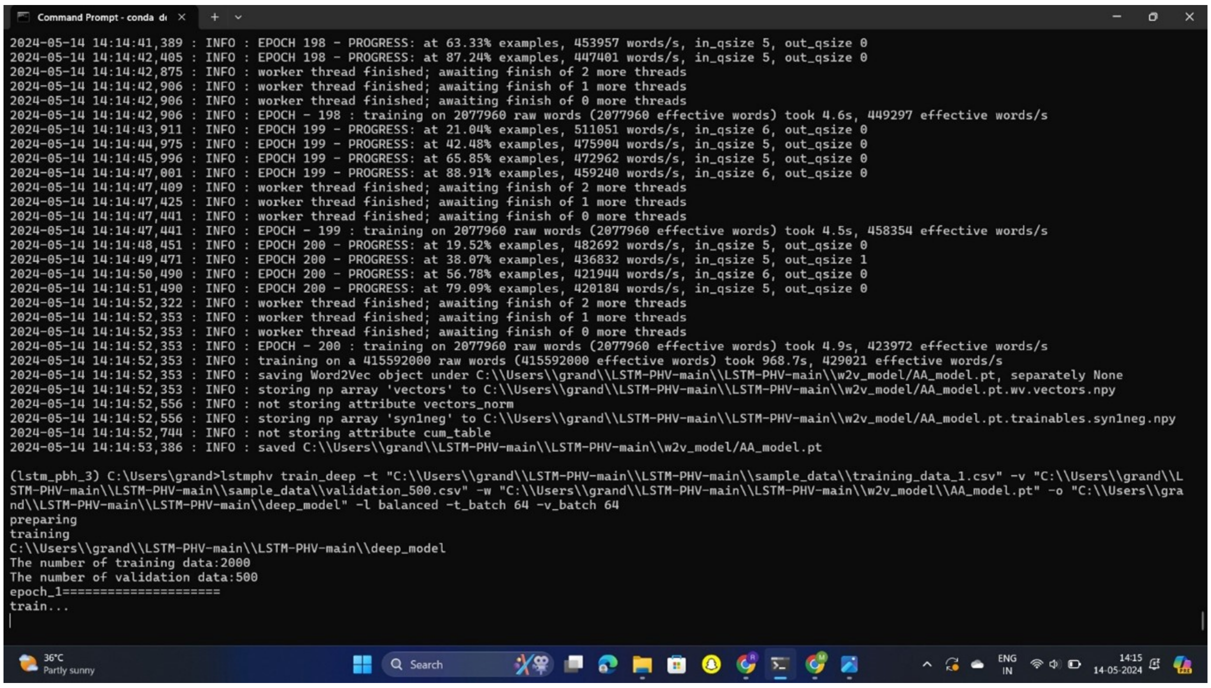
Command 1:

lstmphv train_w2v -i "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\sample_data\\w2v_sample_1.fa" -o "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model"



```
2024-05-14 14:09:06,322 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:09:06,353 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:09:06,369 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:09:06,369 : INFO : EPOCH - 126 : training on 2077960 raw words (2077960 effective words) took 4.6s, 450144 effective words/s
2024-05-14 14:09:07,380 : INFO : EPOCH 127 - PROGRESS: at 20.54% examples, 502924 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:08,407 : INFO : EPOCH 127 - PROGRESS: at 40.20% examples, 460454 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:09,420 : INFO : EPOCH 127 - PROGRESS: at 61.83% examples, 454559 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:10,414 : INFO : EPOCH 127 - PROGRESS: at 84.59% examples, 445223 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:11,000 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:09:11,016 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:09:11,032 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:09:11,032 : INFO : EPOCH - 127 : training on 2077960 raw words (2077960 effective words) took 4.6s, 447192 effective words/s
2024-05-14 14:09:12,045 : INFO : EPOCH 128 - PROGRESS: at 20.54% examples, 499969 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:13,060 : INFO : EPOCH 128 - PROGRESS: at 40.59% examples, 465977 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:14,088 : INFO : EPOCH 128 - PROGRESS: at 62.89% examples, 458160 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:15,104 : INFO : EPOCH 128 - PROGRESS: at 86.67% examples, 451130 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:15,607 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:09:15,623 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:09:15,623 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:09:15,623 : INFO : EPOCH - 128 : training on 2077960 raw words (2077960 effective words) took 4.6s, 452374 effective words/s
2024-05-14 14:09:16,633 : INFO : EPOCH 129 - PROGRESS: at 12.59% examples, 328744 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:17,648 : INFO : EPOCH 129 - PROGRESS: at 32.69% examples, 381136 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:18,658 : INFO : EPOCH 129 - PROGRESS: at 49.41% examples, 375465 words/s, in_qsize 6, out_qsize 1
2024-05-14 14:09:19,670 : INFO : EPOCH 129 - PROGRESS: at 72.89% examples, 391788 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:20,713 : INFO : EPOCH 129 - PROGRESS: at 95.61% examples, 393990 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:20,840 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:09:20,856 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:09:20,860 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:09:20,860 : INFO : EPOCH - 129 : training on 2077960 raw words (2077960 effective words) took 5.2s, 397886 effective words/s
2024-05-14 14:09:21,933 : INFO : EPOCH 130 - PROGRESS: at 19.52% examples, 449592 words/s, in_qsize 4, out_qsize 2
2024-05-14 14:09:22,960 : INFO : EPOCH 130 - PROGRESS: at 39.43% examples, 435659 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:23,970 : INFO : EPOCH 130 - PROGRESS: at 59.94% examples, 430828 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:25,018 : INFO : EPOCH 130 - PROGRESS: at 82.31% examples, 423395 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:25,711 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:09:25,711 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:09:25,711 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:09:25,711 : INFO : EPOCH - 130 : training on 2077960 raw words (2077960 effective words) took 4.9s, 428187 effective words/s
2024-05-14 14:09:26,726 : INFO : EPOCH 131 - PROGRESS: at 19.52% examples, 480600 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:09:27,750 : INFO : EPOCH 131 - PROGRESS: at 39.83% examples, 455330 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:09:28,775 : INFO : EPOCH 131 - PROGRESS: at 62.35% examples, 455405 words/s, in_qsize 5, out_qsize 0
```

Figure 7.1:



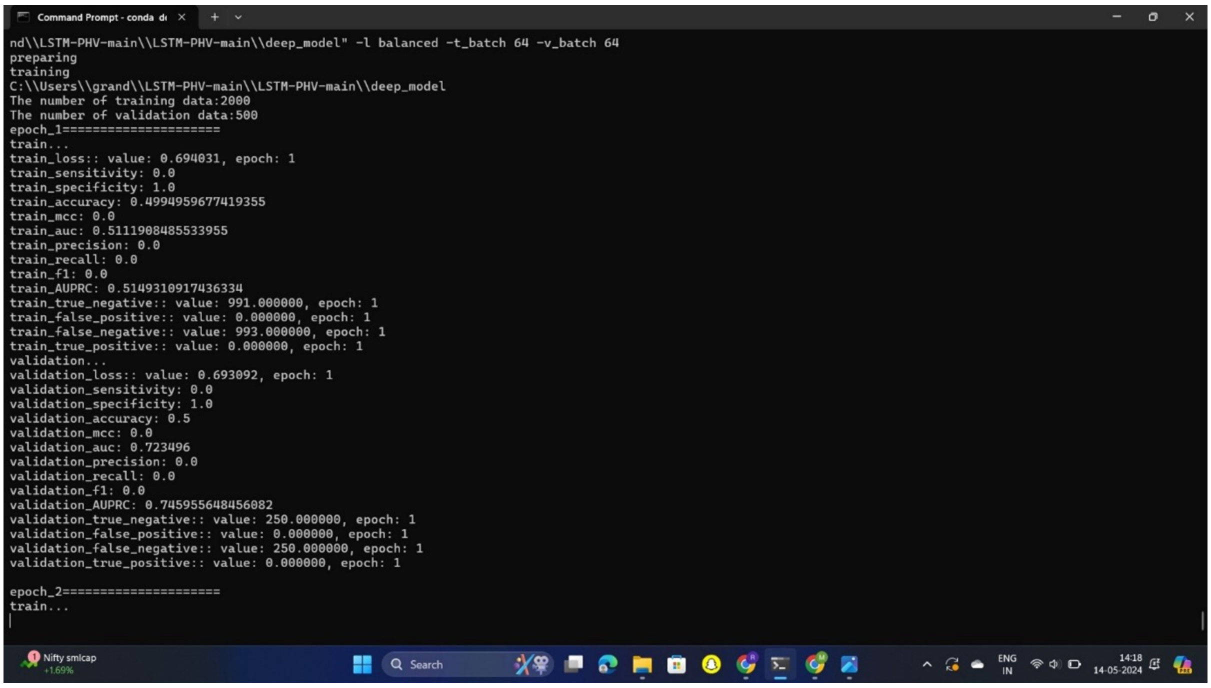
```
2024-05-14 14:14:41,389 : INFO : EPOCH 198 - PROGRESS: at 63.33% examples, 453957 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:14:42,405 : INFO : EPOCH 198 - PROGRESS: at 87.24% examples, 447401 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:14:42,875 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:14:42,906 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:14:42,906 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:14:42,906 : INFO : EPOCH - 198 : training on 2077960 raw words (2077960 effective words) took 4.6s, 449297 effective words/s
2024-05-14 14:14:43,911 : INFO : EPOCH 199 - PROGRESS: at 21.04% examples, 511051 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:14:44,975 : INFO : EPOCH 199 - PROGRESS: at 42.48% examples, 475904 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:14:45,996 : INFO : EPOCH 199 - PROGRESS: at 65.85% examples, 472962 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:14:47,001 : INFO : EPOCH 199 - PROGRESS: at 88.91% examples, 459240 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:14:47,409 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:14:47,425 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:14:47,441 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:14:47,441 : INFO : EPOCH - 199 : training on 2077960 raw words (2077960 effective words) took 4.5s, 458354 effective words/s
2024-05-14 14:14:48,451 : INFO : EPOCH 200 - PROGRESS: at 19.52% examples, 482692 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:14:49,471 : INFO : EPOCH 200 - PROGRESS: at 38.07% examples, 436832 words/s, in_qsize 5, out_qsize 1
2024-05-14 14:14:50,490 : INFO : EPOCH 200 - PROGRESS: at 56.78% examples, 421944 words/s, in_qsize 6, out_qsize 0
2024-05-14 14:14:51,490 : INFO : EPOCH 200 - PROGRESS: at 79.09% examples, 420184 words/s, in_qsize 5, out_qsize 0
2024-05-14 14:14:52,322 : INFO : worker thread finished; awaiting finish of 2 more threads
2024-05-14 14:14:52,353 : INFO : worker thread finished; awaiting finish of 1 more threads
2024-05-14 14:14:52,353 : INFO : worker thread finished; awaiting finish of 0 more threads
2024-05-14 14:14:52,353 : INFO : EPOCH - 200 : training on 2077960 raw words (2077960 effective words) took 4.9s, 423972 effective words/s
2024-05-14 14:14:52,353 : INFO : training on a 415592000 raw words (415592000 effective words) took 968.7s, 429021 effective words/s
2024-05-14 14:14:52,353 : INFO : saving Word2Vec object under C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model/AA_model.pt, separately None
2024-05-14 14:14:52,353 : INFO : storing np array 'vectors' to C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model/AA_model.pt.wv.vectors.npy
2024-05-14 14:14:52,556 : INFO : not storing attribute vectors_norm
2024-05-14 14:14:52,556 : INFO : storing np array 'syn1neg' to C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model/AA_model.pt.trainables.syn1neg.npy
2024-05-14 14:14:52,744 : INFO : not storing attribute cum_table
2024-05-14 14:14:53,386 : INFO : saved C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model/AA_model.pt

(lstm_pbh_3) C:\Users\grand>lstmphv train_deep -t "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\sample_data\\training_data_1.csv" -v "C:\\Users\\grand\\L
STM-PHV-main\\LSTM-PHV-main\\sample_data\\validation_500.csv" -w "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model\\AA_model.pt" -o "C:\\Users\\gra
nd\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model" -l balanced -t_batch 64 -v_batch 64
preparing
training
C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model
The number of training data:2000
The number of validation data:500
epoch_1====================
train...
```

Figure 7.2:

Command 2:

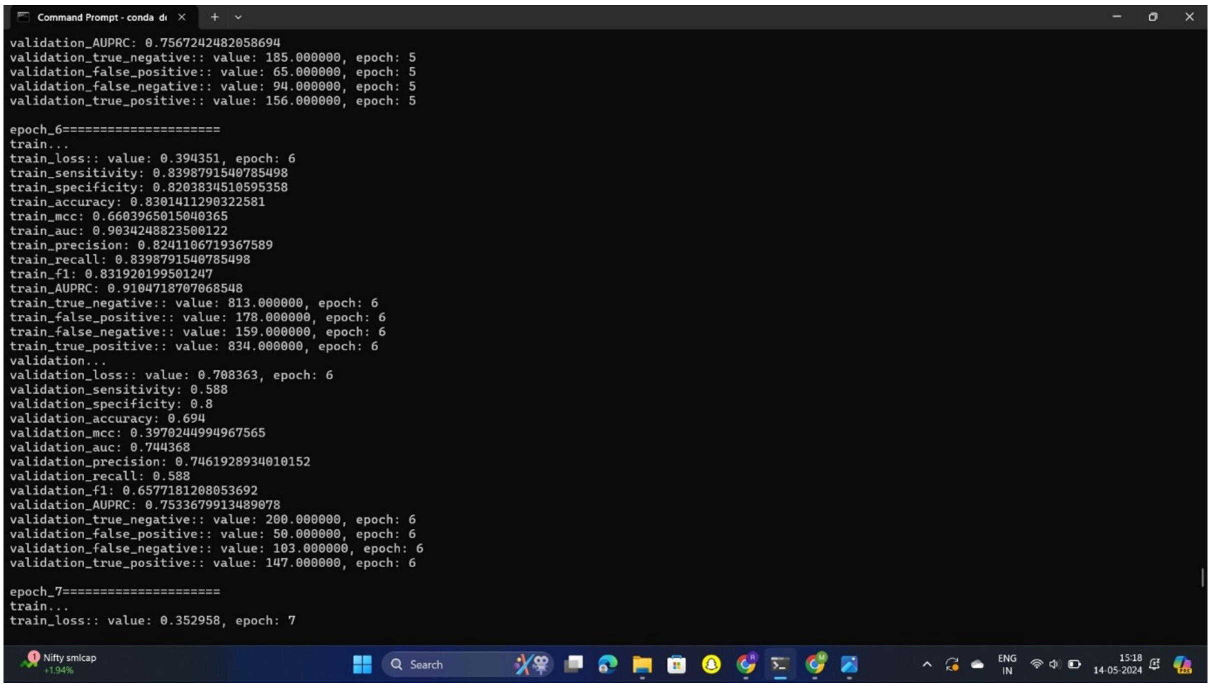
lstmphv train_deep -t ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV main\\sample_data\\training_data_1.csv” -v ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\sample_data\\validation_500.csv” -w ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model\\AA_model.pt” -o ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model” -l balanced -t_batch 64 -v_batch 64

The above command trains the LSTM-PHV model,using the pre-trained Word2Vec model (AA_model.pt), the training and validation datasets (training_data_1.csv and validation_500.csv), and the trained model saved to the designated output directory (deep_model). The batch sizes for training and validation are specified by the flags -t_batch 64 and -v_batch 64, respectively, while the -l balanced flag specifies that the training data is balanced to efficiently train the LSTM-PHV model for protein-protein interaction prediction.



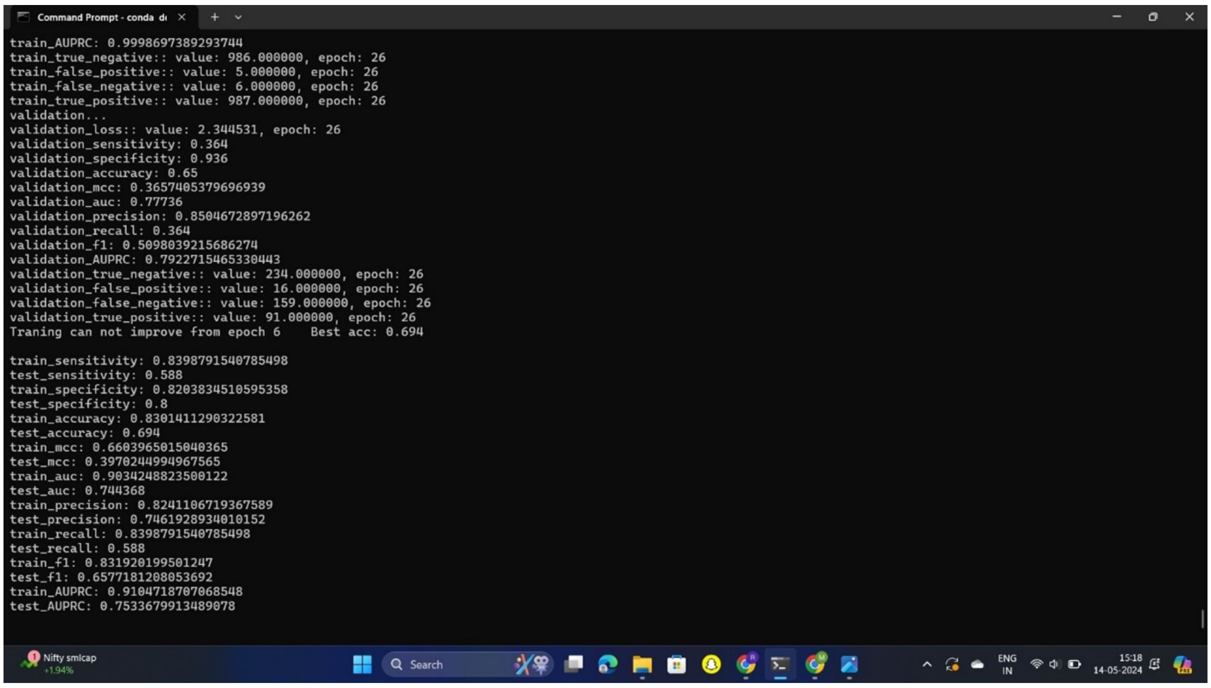
```
nd\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model" -l balanced -t_batch 64 -v_batch 64
preparing
training
C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model
The number of training data:2000
The number of validation data:500
epoch_1====================
train...
train_loss:: value: 0.694031, epoch: 1
train_sensitivity: 0.0
train_specificity: 1.0
train_accuracy: 0.4994959677419355
train_mcc: 0.0
train_auc: 0.5111908485533955
train_precision: 0.0
train_recall: 0.0
train_f1: 0.0
train_AUPRC: 0.5149310917436334
train_true_negative:: value: 991.000000, epoch: 1
train_false_positive:: value: 0.000000, epoch: 1
train_false_negative:: value: 993.000000, epoch: 1
train_true_positive:: value: 0.000000, epoch: 1
validation...
validation_loss:: value: 0.693092, epoch: 1
validation_sensitivity: 0.0
validation_specificity: 1.0
validation_accuracy: 0.5
validation_mcc: 0.0
validation_auc: 0.723496
validation_precision: 0.0
validation_recall: 0.0
validation_f1: 0.0
validation_AUPRC: 0.745955648456082
validation_true_negative:: value: 250.000000, epoch: 1
validation_false_positive:: value: 0.000000, epoch: 1
validation_false_negative:: value: 250.000000, epoch: 1
validation_true_positive:: value: 0.000000, epoch: 1

epoch_2====================
train...
```

Figure 7.3:



```
validation_AUPRC: 0.7567242482058694
validation_true_negative:: value: 185.000000, epoch: 5
validation_false_positive:: value: 65.000000, epoch: 5
validation_false_negative:: value: 94.000000, epoch: 5
validation_true_positive:: value: 156.000000, epoch: 5

epoch_6====================
train...
train_loss:: value: 0.394351, epoch: 6
train_sensitivity: 0.8398791540785498
train_specificity: 0.8203834510595358
train_accuracy: 0.8301411290322581
train_mcc: 0.6603965015040365
train_auc: 0.9034248823500122
train_precision: 0.8241106719367589
train_recall: 0.8398791540785498
train_f1: 0.831920199501247
train_AUPRC: 0.9104718707068548
train_true_negative:: value: 813.000000, epoch: 6
train_false_positive:: value: 178.000000, epoch: 6
train_false_negative:: value: 159.000000, epoch: 6
train_true_positive:: value: 834.000000, epoch: 6
validation...
validation_loss:: value: 0.708363, epoch: 6
validation_sensitivity: 0.588
validation_specificity: 0.8
validation_accuracy: 0.694
validation_mcc: 0.3970244994967565
validation_auc: 0.744368
validation_precision: 0.7461928934010152
validation_recall: 0.588
validation_f1: 0.6577181208053692
validation_AUPRC: 0.7533679913489078
validation_true_negative:: value: 200.000000, epoch: 6
validation_false_positive:: value: 50.000000, epoch: 6
validation_false_negative:: value: 103.000000, epoch: 6
validation_true_positive:: value: 147.000000, epoch: 6

epoch_7====================
train...
train_loss:: value: 0.352958, epoch: 7
```

Figure 7.4:



```
train_AUPRC: 0.9998697389293744
train_true_negative:: value: 986.000000, epoch: 26
train_false_positive:: value: 5.000000, epoch: 26
train_false_negative:: value: 6.000000, epoch: 26
train_true_positive:: value: 987.000000, epoch: 26
validation...
validation_loss:: value: 2.344531, epoch: 26
validation_sensitivity: 0.364
validation_specificity: 0.936
validation_accuracy: 0.65
validation_mcc: 0.3657405379696939
validation_auc: 0.77736
validation_precision: 0.8504672897196262
validation_recall: 0.364
validation_f1: 0.5098039215686274
validation_AUPRC: 0.7922715465330443
validation_true_negative:: value: 234.000000, epoch: 26
validation_false_positive:: value: 16.000000, epoch: 26
validation_false_negative:: value: 159.000000, epoch: 26
validation_true_positive:: value: 91.000000, epoch: 26
Traning can not improve from epoch 6    Best acc: 0.694

train_sensitivity: 0.8398791540785498
test_sensitivity: 0.588
train_specificity: 0.8203834510595358
test_specificity: 0.8
train_accuracy: 0.8301411290322581
test_accuracy: 0.694
train_mcc: 0.6603965015040365
test_mcc: 0.3970244994967565
train_auc: 0.9034248823500122
test_auc: 0.744368
train_precision: 0.8241106719367589
test_precision: 0.7461928934010152
train_recall: 0.8398791540785498
test_recall: 0.588
train_f1: 0.831920199501247
test_f1: 0.6577181208053692
train_AUPRC: 0.9104718707068548
test_AUPRC: 0.7533679913489078
```

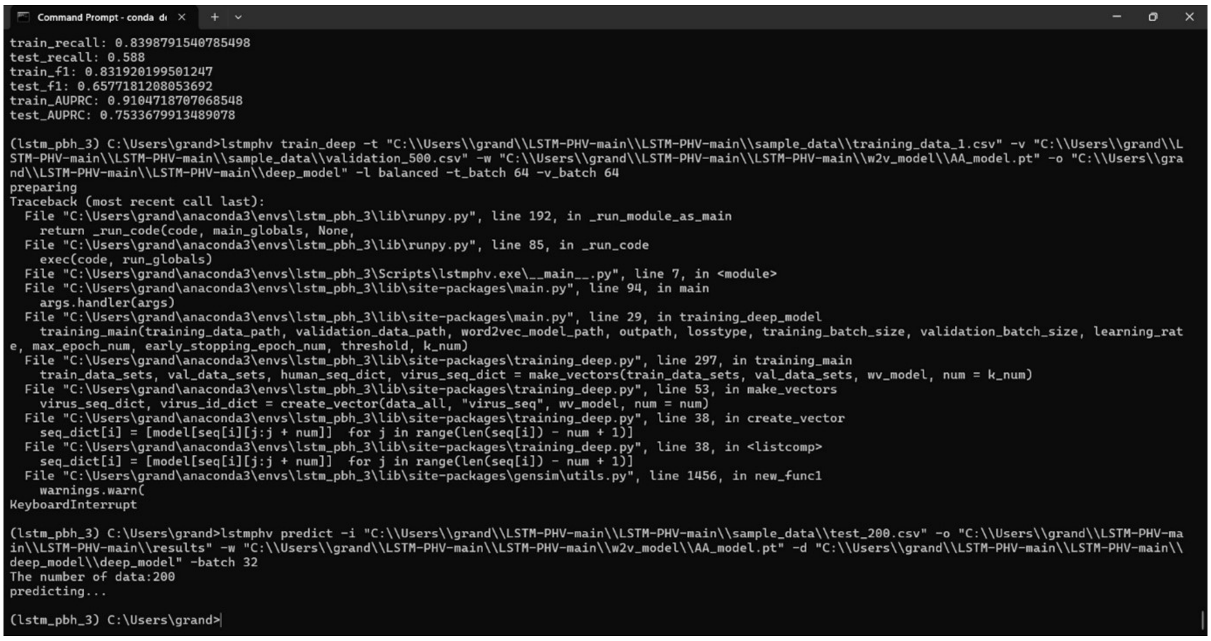
Figure 7.5:

As we can see that we got the best epoch at 6 and we cannot get the best accuracy after that.

Command 3:

lstmphv predict -i ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\sample_data\\test_200.csv” -o ”C:\\Users\\grand\\LSTM-PHV-main \\LSTM-PHV-main\\results” -w ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model\\AA_model.pt” -d ”C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model\\deep_model” -batch 32

Using the LSTM-PHV model, this command forecasts Protein-Protein Interactions (PPI). Its input consists of a pre-trained Word2Vec model (AA_model.pt), a trained LSTM-PHV model (deep_model), and a test dataset (test_200.csv). With a batch size of 32, the predictions are recorded to the designated output directory (results).



```
train_recall: 0.8398791540785498
test_recall: 0.588
train_f1: 0.831920199501247
test_f1: 0.6577181208053692
train_AUPRC: 0.9104718707068548
test_AUPRC: 0.7533679913489078

(lstm_pbh_3) C:\Users\grand>lstmphv train_deep -t "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\sample_data\\training_data_1.csv" -v "C:\\Users\\grand\\L
STM-PHV-main\\LSTM-PHV-main\\sample_data\\validation_500.csv" -w "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model\\AA_model.pt" -o "C:\\Users\\gra
nd\\LSTM-PHV-main\\LSTM-PHV-main\\deep_model" -l balanced -t_batch 64 -v_batch 64
preparing
Traceback (most recent call last):
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\runpy.py", line 192, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\Scripts\lstmphv.exe\__main__.py", line 7, in <module>
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\main.py", line 94, in main
    args.handler(args)
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\main.py", line 29, in training_deep_model
    training_main(training_data_path, validation_data_path, word2vec_model_path, outpath, losstype, training_batch_size, validation_batch_size, learning_rat
e, max_epoch_num, early_stopping_epoch_num, threshold, k_num)
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\training_deep.py", line 297, in training_main
    train_data_sets, val_data_sets, human_seq_dict, virus_seq_dict = make_vectors(train_data_sets, val_data_sets, wv_model, num = k_num)
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\training_deep.py", line 53, in make_vectors
    virus_seq_dict, virus_id_dict = create_vector(data_all, "virus_seq", wv_model, num = num)
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\training_deep.py", line 38, in create_vector
    seq_dict[i] = [model[seq[i][j:j + num]]  for j in range(len(seq[i]) - num + 1)]
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\training_deep.py", line 38, in <listcomp>
    seq_dict[i] = [model[seq[i][j:j + num]]  for j in range(len(seq[i]) - num + 1)]
  File "C:\Users\grand\anaconda3\envs\lstm_pbh_3\lib\site-packages\gensim\utils.py", line 1456, in new_func1
    warnings.warn(
KeyboardInterrupt

(lstm_pbh_3) C:\Users\grand>lstmphv predict -i "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\sample_data\\test_200.csv" -o "C:\\Users\\grand\\LSTM-PHV-ma
in\\LSTM-PHV-main\\results" -w "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\w2v_model\\AA_model.pt" -d "C:\\Users\\grand\\LSTM-PHV-main\\LSTM-PHV-main\\
deep_model\\deep_model" -batch 32
The number of data:200
predicting...

(lstm_pbh_3) C:\Users\grand>
```

Figure 7.6:

	Train	*Test*
Recall	0.8398791540785498	0.588
Accuracy	0.8301411290322581	0.694
Precision	0.8241106719367589	0.7461928934010152
AUC	0.90342488235001202	0.744368
specificity	0.8203834510595358	0.8
F1	0.831920199501247	0.6577181208653692
AUPRC	0.9104718707868548	0.7533679913489078

Table 7.1: Result

Chapter 8

CONCLUSION

In order to properly predict PPIs between people and viruses, we proposed the LSTM-PHV, which combined LSTM and word2vec embedding approach together by considering the complete sequence context of amino acid residues. Word2vec can be used to preserve the information about the local amino acid residue patterns. Interestingly, our method does not utilize the biological properties of amino acid residues, despite the fact that earlier models mostly depended on them. The LSTM learns more information about the amino acid patterns across the entirety of the sequence. The LSTM-PHV learned incredibly imbalanced data and was able to predict the binding of a human protein to an unknown viral protein with high reliability, in contrast to existing state-of-the-art models. With an accuracy rate of 83 percent and an AUPRC of 0.91, the model shows promise in catching significant interactions. The model's strong recall score of 0.84 suggests that it can accurately detect a significant percentage of actual positive interactions. Specificity (82 percentage) and F1 score (0.83), on the other hand, both show potential for improvement and point to the model's propensity to misclassify some genuine negatives as false positives. Overall, our findings demonstrate the potential of the LSTM-PHV model for precise PPI prediction, with room for improvement in terms of specificity and F1 score.

8.1 SCOPE OF FURTHER WORK

Increasing the quantity of training samples will greatly increase the LSTM-PHV model's learning capacity and accuracy in the project's future work. The model can capture a wider spectrum of protein-protein interactions by growing the dataset, which will result in more reliable predictions. To further aid attain optimal performance, elements like the learning rate may be changed. By adjusting the learning rate precisely, the model may be made to discover the most accurate answers by preventing it from converging too rapidly or too slowly. Furthermore, it is possible to expand the current LSTM-PHV model to forecast interactions between additional viruses, including the proteins of the dengue and human viruses. Through the application of this model to various viral protein sequences, scientists can learn more about the mechanisms behind viral infection and possibly discover new targets for therapeutic intervention. This extension of the model's use can assist in the creation of antiviral tactics and further our understanding of viral pathogenesis.

All things considered, further research utilizing LSTM networks for PPI prediction has enormous potential to further our comprehension of protein interactions and their function in biological processes. Researchers may create more precise and understandable models that offer important insights into the intricate realm of protein interactions by focusing on these important areas.

REFERENCES

[1] **Berg J.M** et al. (2002), ”Biochemistry”, 5th Edition, (W. H. Freeman and Company), ISBN:9780716746843.

[2] **Alberts, B.** et al. (2002), ”Molecular Biology of the Cell”, 4th Edition, (Garland Science), ISBN:9780815332183.

[3] **Brown, C.R.** et al. (2015), ”Chromatin structure analysis of single gene molecules by psoralen cross-linking and electron microscopy”, Methods in Molecular Biology, Vol. 1228, pp. 93–121, (Springer), DOI: http://doi.org/10.1007/978-1-4939-1652-8_7.

[4] **Spanos, C.** et al. (2018), ”Proteome Science”, Proteome Science, 16(5), DOI: http://doi.org/10.1186/s12953-018-0134-0

[5] **Jones G.E.** (2010), ”Moving into the third Dimension”, 87(12), Nature Reviews Molecular Cell Biology, 11(231), DOI:http://doi.org/10.1038/nrm2872.

[6] **Lepore R.** et al.(2019) ”RNA target highlights in CASP15: Evaluation of predicted models by structure providers”, Proteins: Structure, Function, and Bioinformatics, 87(12), DOI: https://doi.org/10.1002/prot.26550

[7] **Ahmed I.** et al.(2018), ”Prediction of human-bacillus anthracis protein–protein interactions using multi-layer neural network.”, Bioinformatics, 34(24):4159-4164, DOI: http://doi.org/10.1007/978-1-4939-1680-1_9.

[8] **Tsukiyama S.** et al.(2021), ”LSTM-PHV: Prediction of human-virus protein-protein interactions by LSTM with word2vec”, Briefings in Bioinformatics, 22(6), DOI: https://doi.org/10.1093/bib/bbab228

[9] **Zhou H.** et al.(2019), ”Improving neural protein-protein interaction extraction with knowledge selection”, Comput Biol Chem , 83:107146, DOI: 10.1016/j.compbiolchem.2019.107146

[10] **Hu X.** et al.(2022), ”Deep learning frameworks for protein–protein interaction prediction.”, Comput Struct Biotechnol J ., DOI: http://doi.org/10.1016/j.csbj.2022

[11] **Baranwal M.** et al.(2022), ”Struct2Graph: a graph attention network for structure based predictions of protein–protein interactions”, BMC Bioinformatics , 23(1), 370. DOI: http://doi.org/10.1186/s12859-022-04525-x

[12] **Mikolov T.** et al.(2013), ”Distributed Representations of Words and Phrases and their Compositionality”, Advances in Neural Information Processing Systems 26 (NIPS 2013)., DOI: https://doi.org/10.48550/arXiv.1310.4546