Protein protein interaction prediction using Deep Learning-

a critical review and development of a novel DenseNet based prediction strategy

A thesis submitted in partial fulfillment of the requirement for the degree of

Master of Engineering

in

Computer Science and Engineering

Submitted by

Aanzil Akram Halsana

Registration No.: 154152 of 2020-2021,

Examination Roll No.: M4CSE22028

Session: 2020-2022

Under the Supervision of

Prof. Subhadip Basu

Department of Computer Science and Engineering

Jadavpur University,

188, Raja S.C. Mallick Rd,

Kolkata - 700032,

West Bengal, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that this is a bonafide record of the project entitled **"Protein protein interaction prediction using deep learning - a critical review and development of a novel DenseNet based prediction strategy"**, submitted by Aanzil Akram Halsana (University Registration No.: 154152 of 2020-2021 , Examination Roll No.: M4CSE22028) is hereby approved of a creditable study of a technological subject carried out under my supervision and presented in a manner satisfactory to warrant its acceptance for partial fulfillment of the requirements of the degree of Master of Engineering in Computer Science and Engineering. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other university or institute.

Supervisor

. . . . . .

Prof. Subhadip Basu
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

Countersigned

. . . . . .

Prof. Anupam Sinha
Head, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

. . . . . .

Prof. Chandan Mazumdar
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata-32, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval[1]

This is to certify that the thesis entitled **"Protein protein interaction prediction using deep learning - a critical review and development of a novel DenseNet based prediction strategy"**, is a bonafide record of work carried out by Aanzil Akram Halsana in partial fulfillment of the requirements of the degree of Master of Engineering in Computer Science and Engineering in Department of Computer Science and Engineering, Jadavpur University during the period of June 2021 to June 2022. It is understood that by this approval the undersigned do not necessarily endorse any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

. . . . . .
Signature of Examiner 1
Date :

. . . . . .
Signature of Examiner 2
Date :

[1]Only in case thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled **"Protein protein interaction prediction using deep learning - a critical review and development of a novel DenseNet based prediction strategy"** contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Engineering in Computer Science and Engineering.

All information has been obtained and presented in accordance with academic rules an ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name : Aanzil Akram Halsana

Registration No.: 154152 of 2020-2021

Examination Roll No.: M4CSE22028

Thesis Title : Protein protein interaction prediction using deep learning - a critical review and development of a novel DenseNet based prediction strategy

.....

Signature with date

Acknowledgement

The success and final outcome of this thesis required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I owe my deep gratitude to my mentor and thesis guide **Prof. Subhadip Basu**, who took keen interest on my thesis work and guided me all along, till the completion of my thesis work by providing all the necessary information for developing a good end-to-end solution. I am extremely thankful to him for his assistance and dedicated involvement in every step throughout the process, despite the devastating COVID-19 epidemiological situation. His constant suggestions and editorial enhancements has been an everlasting source of motivation and inspiration to me. It has been an honour to work under a person with eternal wisdom yet kind, simplistic attitude.

I would like to thank **Prof. Mita Nasipuri**, former co-ordinator and **Prof. Mahantapas Kundu**, present co-ordinator of Centre for Microprocessor Application for Training and Educational Research (CMATER) lab, Department of Computer Science and Engineering, Jadavpur University for providing me all necessary infrastructure and technical facility t carry out this research.

I would also like to thank **Prof. Anupam Sinha**, Head of the department, Department of Computer Science and Engineering, Jadavpur University for extending his help to complete this thesis.

Besides my supervisor, I would also like to thank **Dr. Anup Kr. Halder** and **Mr. Tapas Chakraborty**, for their constant support, encouragement, insightful comments, and research expertise which helped me the most. Without their enthusiasm, support and continuous optimism, this thesis wouldn't have progressed much.

Most importantly, none of this could have happened without the love and support of my family. To my father **Mr. Hazrat Ali Halsana**, my mother **Mrs. Rehena Halsana** and my brother **Mr. Atique Bikram Halsana** – it would be an understatement to say that, their unconditional love and encouragement has always helped me in my need. With their forbearance and whole-hearted support, this thesis would not have been able to see the light of the day.

Finally, I would like to thank all my Professors of Jadavpur University, my classmates and the many well-wishers who have been constant source of help and inspiration to me during my thesis work. My thanks and appreciation to all of them for being a part of this wonderful journey and making this possible.

.....

Aanzil Akram Halsana
Examination Roll No.: M4CSE22028
Dept. of Computer Science & Engineering
Jadavpur University
Kolkata, India

Abstract

Protein-protein interactions(PPI) are crucial for understanding behaviour of living organisms and identifying causes of diseases. In this thesis, a novel image based deep learning method has been proposed for predicting PPI, which we call DensePPI. Novelty of our approach is that we represent PPI using images. PPI images were generated based on sequences of amino acids present in the two interacting proteins. Various colours have been used to represent each Amino Acid. Rectangular images so generated were further divided into square sub-images using horizontal and vertical strides. Square sub-images were then given as input to a Deep learning model (DenseNet201) for automatic feature extraction and prediction. A consensus-based model was also introduced to tackle the problem enormous data size and high model complexity. Model was trained using the Pan *et al.*'s dataset and *S.Cerevisiae* datset. Model's performance was tested on independent datasets like *Caenorhabditis elegans*, *Escherichia coli*, *Helicobacter Pylori*, *Homo sapiens* and *Mus Musculus* PPI after removing sequence similarities. Maximum accuracies on those datasets were 99.95%, 100.00%, 99.90%, 99.90% and 100.00% respectively. The Consensus model achieved a high accuracy of 98.86% for external test sets. Improved performance of DensePPI shows that the image based DL classifier could be effective for PPI prediction. DensePPI can be used to predict cross-species interactions, based on the enhanced prediction accuracies obtained on separate test sets. It can also give researchers new insights into signalling pathway analysis, therapeutic target prediction, and disease pathophysiology.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Protein is a necessary component of all living organisms and is involved in a variety of processes including metabolism, signal transmission, hormone control, DNA transcription, and replication. Proteins, in general, interact with other proteins to fulfil their activities in complexes [1]. The systematic mapping of physical protein-protein interactions (PPIs) in cells has proven immensely useful in furthering our knowledge of protein function and biology. This PPI network information has proven useful for downstream inference tasks in understanding functional genomics and biological pathway analysis in species such as yeast and humans where a substantial network of experimentally identified PPIs exists [2]. Understanding cellular biological processes in normal and pathological states requires precise detection of PPIs and identifying the interaction types. The findings of these investigations may aid in the discovery of medicinal targets [3].

However, because biological experiment methods are expensive and time-consuming, protein interactions found using experimental methods can only account for a small portion of the total PPIs networks. Furthermore, the detection results are affected by the experimental environment and operational processes, which might lead to false positives and negatives. As a result, establishing trustworthy computational methods for precisely predicting protein interactions is critical.

The recent improvements in computational power and advancement of big data has contributed in transforming large quantity data into valuable knowledge. Massive amount of data in the field of medical science and bioinformatics including image, signal and omics data has been accumulated, with the help of engineering and data analysis methods, it shows potential for industrial application and academic discipline [4].

Many computational and statistical methods have recently been developed to overcome this challenge. Some have attempted to mine new protein data, while others have entailed the development of new machine learning algorithms [5]. This paper briefly reviews the key terminologies and the previous machine learning based works followed by the progress that we have made on predicting the protein protein interaction computationally.

1.1 Some key ideas

This section describes the biological background for proteins and it's interaction.

1.1.1 Protein–protein interaction definition

Protein–protein interactions (PPIs) are highly specialised physical contacts formed between two or more protein molecules as a result of biochemical activities guided by electrostatic forces, hydrogen bonding, and the hydrophobic effect. Many are physical interactions between chains that take place in a cell or in a living organism in a specific biomolecular setting.

1.1.2 Biological background

The proteome is the collection of all proteins expressed by a cell or organism. The majority of these proteins do not function alone. Instead, they rely on interactions with a large number of other proteins to carry out their roles in the cell.

Signal transduction, cell-cell contact, transport, metabolism, cell motility, and even antigen-antibody identification are all biological functions that rely on protein-protein interactions. The protein interactome is a term that refers to the entire set of protein–protein interactions that exist in a biological system (PPIs).

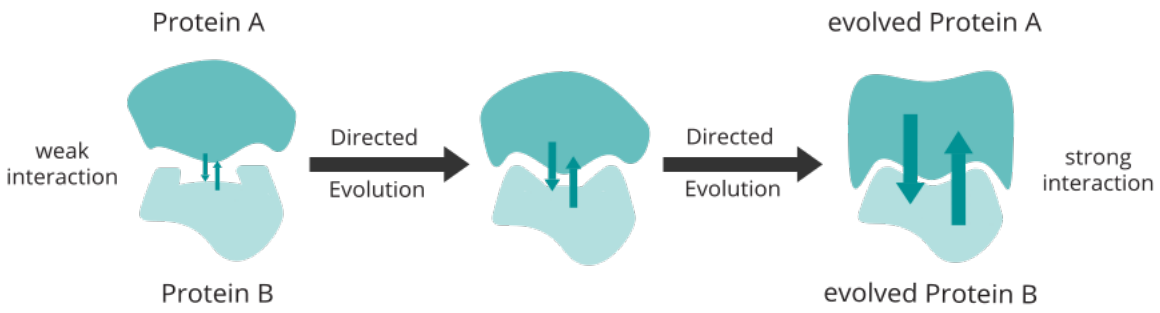

Figure 1.1: PPIs in evolution context. Image courtesy : team Heidelberg, IGEM, 2017.

1.2 Motivation

In order to solve the problem of PPI prediction computationally, we first studied related works with computational PPI prediction and then tried to find the research gaps. It was found that several extracted features from PPI has been in consideration for long time. However, none of them explicitly worked by giving equal importance to each and every region of protein protein interaction. We studied an *state-of-the-art* image classifier, so we wanted to convert a PPI to an image so that we can classify it accurately. We tried learning the underlying interaction between by providing equal probability to protein sequence interaction segments.

In this thesis, we propose **DensePPI**, a novel approach for prediction of PPIs using an Deep Learning (DL) based predictor by translating the protein interactions into images. This method ensures equality of prediction capability for every region in the AA residues of the proteins participating in an interaction. AA sequences of query interacting protein pairs were converted into images followed by generation of sub-images from each of them. These sub-images were used to measure performance of the overall framework. Our DL method has acquired performance greater than that of the *state-of-the-art* models on standard human dataset and other interspecies datasets.

1.3 Organization of thesis

This section describes the overall outline of the entire thesis. The remaining chapters of the thesis are organised as follows. Chapter 2 discusses the available methodologies, general strategies, synopsis of deep learning architectures, *state-of-the-art* method and

their comparison for predicting PPIs. Chapter 3 is primarily concerned with our proposed methodology, rigorous experimentations, PPI datasets used and the metrics to judge our proposed methodology. The results obtained from our approach discussed in chapter 3 are discussed in chapter 4. Performance comparison, experimental findings are reported in chapter 4. Finally, this thesis ends with chapter 5 which concludes this thesis by summarizing our methodology, the limitations that our approach has and it also sheds light on future scope of our work.

CHAPTER 2

Literature Review

In the research process, the literature review plays a critical role. It's a place where researchers get their research ideas, which are then refined into concepts and finally theories. It also gives the researcher a bird's eye view of previous research in the field. A researcher will know where his or her research stands based on the findings of the literature review. This chapter includes the literature survey of the methodologies used for prediction of protein protein interactions (PPIs).

2.1 General Strategy

Before beginning a new investigation, a literature review builds familiarity with and comprehension of current research in a certain topic. We should be able to find out what research has already been done and discover what is unknown about our topic by conducting a literature review. Our main focus was on Deep Learning(DL) architectures which are present for Protein Protein Interaction (PPI) prediction. Thus, the upcoming review is on DL methods present for PPI prediction.

DL technology has recently risen to prominence as a result of various scientific studies that aid in a variety of applications such as image recognition, speech recognition,

machine language translation, computer vision, and many more. Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs), in particular, have made significant contributions to real-world applications and have made human efforts easier. In the realm of bioinformatics, a number of notable DL-based studies have been published.

2.1.1 Outline of Deep Networks

DL architectures are ANNs with several layers, and researchers have contributed a variety of DL architectures dependent on the input and goal of the study. This review focuses on three types of deep learning architectures: DNNs, CNNs, and RNNs. Several researchers, on the other hand, integrated all DL architectures in DNNs [6, 7]. This work looks at 'DNNs' to talk about SAE [5], which uses AEs [8] as the basic unit of NNs [9]. The basis for these concerns is the paper's narrow scope, which focuses on delivering the relevance of DNs utilising sequential information from PPI's input data for the prediction task. In general, there are two main features in DL architectures that improve performance: optimization and regularisation.

During training, the goal is to adjust the weight parameters in each layer so that the key and relevant properties of the input may be learned by filtering out the irrelevant data and transferring an abstract form or reduced number of features to the next layer.. The weight parameters are updated using an algorithm based on the SGD [10] during the optimization operation. Regularization is a technique for avoiding the overfitting problem that happens frequently during training. Weight decay [11], Dropout [12], and rnnDrop [13] are some of the regularisation processes that have been developed. A unique regularisation technique has just been presented [14], which runs in batches and does feature normalisation.

The next section provides a brief overview of three deep learning (DL) approaches: DNNs, RNNs, and CNNs, all of which have made significant contributions to the prediction task of PPIs utilising sequential information alone.

Deep Neural Networks

In simple terms, a DNN is a deep network that has several hidden layers in addition to the input and output layers. The outputs are calculated progressively with the layers of the network for the provided input data. The output of the preceding layers' unit is

included in the input vector at each layer, which is then multiplied by the weight vector of the considered layer to get the weighted total. The output of a layer is computed by applying a non-linear function (ReLU, sigmoid, etc.) [15] to the weighted sum, resulting in more abstract representations of the previous layer output as follows:

$$f_x^{(O+1)} = \mu(w^{(O+1)} f_x^O + z_x^{(O+1)}) \tag{2.1}$$

where μ represents activation, w is the weight matrix, f^O is the inputted data for the O^{th} layer and z is the bias term.

DNNs are excellent at analysing high-dimensional data. Because good bioinformatics research cannot be performed with limited data, the data accessible in this discipline is typically high-dimensional and complicated, and DNNs ensure that researchers have favourable working conditions. By extracting highly abstract and connected information from data, DNNs have the ability to make knowledge more easily understandable. Manually created features have frequently been submitted as contributions, despite the fact that raw data is the only prerequisite for DNNs to learn graded features. This suggests that the capabilities of DNNs have not yet been fully exploited. The future improvement of DNNs in bioinformatics is expected to come from investigations into effective methods for encoding unrefined data and extracting reasonable features from it.

Recurrent Neural Networks

RNNs have a recurring link in each hidden layer that is responsible for operating sequential information through some recurrent computation. The previous output (state vector) is stored in the hidden units, and the output is calculated for the present state using the previous state vector and the considered input. The evolution of RNN over time is represented by the following two equations:

$$G_t = \delta(f_t; \theta) \tag{2.2}$$

$$f_t = r(f_{t-1}; X_t; \theta) \tag{2.3}$$

Here, θ represents weights and biases for the network. The first equation expresses the dependency of the output G_t at time t only with the hidden layer f_t by using some computation function, and the second equation shows the dependency of the hidden

layer f_t at time t with that of f_{t-1} at time $t-1$ and the input X_t at time t by using some computation function. RNNs, specifically BRNNs, are widely utilised in applications like speech recognition, Google translator, and others where historical information is necessary for the current output. In terms of the number of layers, RNN structures appear to be simpler than DNN structures, but when the structure of RNN is unrolled over time, it becomes even more complex.

Though this causes two common problems: vanishing gradients and long-term dependencies, researchers have been able to address these problems by incorporating more complicated units and developing RNN variations such as LSTM and GRU. RNNs are now widely used in a variety of fields, including NLP and language interpretation. The nature of determining the PPI is nearly equivalent to the modelling activities carried out in NLP research, as both are aimed at determining the shared impact of two arrangements based on their fundamental characteristics. Proteins are given in more numerous categories, with a broader range of lengths.

Covolutional Neural Networks

CNN is a Deep Learning method that can take an image as input, assign learnable weights and biases to distinct elements of the image, and identify one from the other with the least amount of pre-processing as compared to other classification algorithms. CNN is a feed-forward neural network with neurons that may respond to nearby units in a portion of the coverage and has excellent performance for data feature extraction. Forward propagation is used to calculate the output value, and back propagation is used to alter the weights and biases. The feature map M_i at i^{th} layer is computed as:

$$M_i = f(M_{i-1} \circledast w_i + b_i) \tag{2.4}$$

where w_i is the weight matrix of the i^{th} layer's convolution kernel, bi is the offset vector, f is the activation function, and operator $\circledast$ is the convolution operations operator. The feature map is sampled according to provided criteria by the subsampling layer, which is normally behind the convolutional layer. Several convolution and sub sampling operations are used by the fully connected layer to classify the collected features. The basic mathematical concept behind CNN is that it uses multi-layer data transformation to translate the input matrix M_o to a new feature representation R.

$$R_i = Map(C = c_i | M_o; (w, b)) \tag{2.5}$$

where c_i stands for the i^{th} label class, M_o stands for the input matrix, and R stands for the feature expression. The purpose of CNN training is to reduce the network loss function $R(w, b)$ as much as possible. At the same time, the final loss function $Z(w, b)$ is normally controlled by a norm, and the strength of the overfitting is controlled by the parameter η, to alleviate the overfitting problem.

$$Z(w, b) = R(w, b) + \frac{\eta}{2} w^T w \tag{2.6}$$

The next section shows methods for applying DL approaches to pairwise protein interaction databases.

2.1.2 Prediction using Paired Protein Interaction Dataset

Some studies have demonstrated that DLs are capable of capturing possible features from input protein raw data, while others have combined DLs with hand-crafted features to improve the performance of PPI prediction tasks. As a result, this sub-section is divided into two categories based on whether or not manual feature engineering is used.

Strategy-I: Inclusion of Manually curated Features

The most crucial aspect of developing a computer technique for predicting PPIs is to mine highly preferred traits that can accurately describe proteins. Several papers provided unique approaches for numerically encoding protein information, as shown in Table 2.1, which are widely adopted by several publishers to develop proficient methods for extracting more finely protein interaction information.

Researchers generally use the above techniques to extract features from protein sequences and use them in DL architectures to predict the PPI accurately.

Table 2.1: The reasoning behind several well-liked hand created features used under Strategy I

Features	Perception behind chosen features
AC	Protein characteristics are examined by treating a protein sequence as a collection of signals that are converted into digital form using the appropriate physicochemical properties.
CT	*k-mer* based assembly approach that calculates the frequency of any combination in the entire sequence by grouping together three neighbouring amino acids that appear successively.
LD	Segments of both continuous and discontinuous amino acids can be used to simultaneously extract fine information about protein interaction.
MCD	Uses the interfaces between serially distant but spatially close amino acid residues to effectively cover a variety of underlying continuous and discontinuous portions present in a sequence.
Protein Signature	An strategy to creating signatures that takes into account the length of the amino acid sequence and produces a numerical representation for each protein sequence

Strategy-II: Auto-Feature Engineering based PPI features

Li et al. presented DNN-PPI [16], the first research on sequence-based PPI prediction using DNs that was completely based on auto-feature engineering, i.e. without the addition of manually derived features. The data must be learned by the NN architecture. The input should be in numerical form for the NN architecture to learn the data. As a result, the author assigned each AA a natural number at random and changed the protein sequence accordingly. The embedding layer in the proposed framework captured information about semantic association among AA, three-layered CNNs bagged position-based features of protein sequences, the LSTM layer covered short and longterm dependencies, and the concatenated features were then fed to the FC layer with dropout to identify potential features. Aside from the positive DNN-PPI results, the author also examined the performance by varying the number of CNN layers from 1 to 2 and found no significant differences in accuracy, but faster convergence in loss with the increasing number of layers.

For more accurate results, several scientists have eliminated sequence similarities between the training and testing pairs of proteins. The CD-HIT software [17] is the most commonly used redundancy elimination tool. The CD-HIT software is a large-database incremental clustering approach that is quick and greedy. This was done after a quick word filtering procedure that grouped proteins that were related enough (sequence identity).

Strategy-III: Prediction using Biomedical text Dataset

Hsieh et al. [18] were the first to implement this category in 2017. The author used a bi-directional RNN with an LSTM technique to solve the PPI identification problem. In the scenario, there are three layers to the method: Embedding layer that accepts the protein entities in sentence form and converts each of their words to the corresponding embedding, resulting in a low-dimensional vector with real-values. Essentially, this layer gathered syntactic and semantic data by combining the effects of nearby words. The recurrent layer, namely a Bi-RNN, receives the obtained vector representation. A FC layer takes the contextual and more refined information received by Bi-RNN and uses it to classify PPIs. The author used the 10 fold CV and cross-corpus (CC) testing methods to evaluate the performance using the two largest PPI corpora: a and c, and concluded that DNs are more suitable for extracting rich context information from larger datasets than manual feature engineering, with favourable results in the CV.

In the Bi-LSTM unit, [19] used a stacking method and included an attention layer. The rest of the work and architecture are identical to [19]. Stacked LSTM refers to an LSTM model with multiple hidden layers and many memory units. To capture a high-level abstract demonstration of each word in the sentence, the author used a vertically stacked LSTM. This layer's output is the hidden state representation of its previous layer, which is subsequently used as an input to the attention layer. The attention layer's objective is to provide hints that can be used as a decision factor in interaction information, or to put it another way, it tells how much attention should be paid to a specific word at any given time.

2.2 Classification of PPI Detection Methods

Approaches for detecting protein-protein interactions are divided into three categories: *in vitro*, *in vivo*, and *in silico* methods. A procedure is carried out in a controlled environment outside of a living creature using *in vitro* procedures. Tandem affinity purification, affinity chromatography, coimmunoprecipitation, protein arrays, protein fragment complementation, phage display, X-ray crystallography, and NMR spectroscopy are in vitro approaches for PPI identification. In *in vivo* approaches, a procedure is carried out on the entire living organism. The *in vivo* PPI detection approaches include yeast two-hybrid (Y2H, Y3H) and synthetic lethality. On a computer (or) through computer simulation, *in silico* techniques are used. Sequence-based approaches, structure-based approaches, chromosome proximity, gene fusion, in silico 2 hybrid, mirror tree, phylogenetic tree,

and gene expression-based approaches are all *in silico* methods for PPI discovery [20].

2.3 *State-of-the-art In Silico* Methods for the Prediction of Protein-Protein Interactions.

The yeast two-hybrid (Y2H) method, as well as other *in vitro* and *in vivo* approaches, have resulted in the large-scale development of helpful tools for detecting protein-protein interactions (PPIs) between specified proteins that can occur in various combinations. However, due to the lack of available PPIs, the data provided by these methods may not be reliable. It is preferable to create techniques that predict the whole range of possible interactions between proteins in order to comprehend the total context of prospective interactions between proteins.

To support the interactions discovered by the experimental technique, a range of *in silico* methods have been developed. This survey mainly focusses on *in silico* methods for identification of PPIs.

2.3.1 Sequence-based prediction of PPI using stacked autoencoder

An autoencoder is a type of artificial neural network that uses an unsupervised learning method to infer a function from unlabeled data in order to generate hidden structures. A stacked autoencoder (SAE) is made up of numerous layers of autoencoders that are trained layer by layer. The output of the first layer is linked to the inputs of the next layer. In an article Sun *et al.* used SAEs to study the sequence based PPIs [5].

To code the protein sequences, they employed two methods: one termed the autocovariance technique (AC) and the other called the conjoint triad approach (CT). For coding proteins, the AC technique, which describes how variables at different locations are connected and interact, is commonly employed [21]. Shen *et al.* [22] proposed the CT approach to represent a protein using only its sequencing information. First, the dipole and side chain volumes of all 20 amino acids are grouped into seven categories. The cluster number is then substituted for each amino acid in a protein sequence. Then, from the N-terminus to the C-terminus, a 3-amino acid window is employed to slide across the entire sequence one step at a time. Each protein is represented by a 343-number vector by estimating the frequency of each three-number combination. They used one hidden layer was for both the AC and CT models (protein sequences coded by

AC or CT) for training the SAE and got best results for 400 and 700 neurons respectively for AC and CT models.

This research was the first to apply a deep learning algorithm to sequence-based PPI prediction and achieved good results. They also trained the datasets from other species, such as *E. coli*, *Drosophila*, and *C. elegans* and received promisin results demonstrating its potential in this field.

2.3.2 Predicting PPIs from Matrix-Based Protein Sequence Using CNN and FSRF

Extracting effective feature descriptors is generally established as the key to predicting PPIs. Deep learning belongs to a branch of machine learning. Its objective is to create and simulate the human brain's neural network for learning and data interpretation in a method that mimics the human brain. Deep learning can uncover the laws of data by combining low-level information to construct an abstract high-level representation. Wang *et al.* proposed to employ a deep learning convolutional neural network (CNN) approach to extract hidden valuable information in proteins [1].

By merging CNN with Feature-Selective Rotation Forest (FSRF), they offered CNN-FSRF, a novel technique for predicting PPIs based on protein sequence. The proposed method converts the protein sequence into a Position-Specific Scoring Matrix (PSSM) containing biological evolution information, then uses CNN to objectively and efficiently extract the protein's deeply hidden features, and finally uses FSRF to remove redundant noise information and provide accurate prediction results.

They used Position-Specific Scoring Matrix (PSSM) method that can contain biological evolution information to generate matrix-based numeric descriptors. An element in the PSSM matrix $r_{i,j}$ mean that the probability of the i^{th} residue being mutated into type j of 20 amino acids during the evolutionary process in the protein from multiple sequence alignments using the sequence comparison tool Position-Specific Iterated BLAST (PSI-BLAST). The convolution neural network is a feed-forward neural network. Its neurons can respond to the surrounding units in a part of the coverage and have excellent performance for data feature extraction. Gradient descent method along with a learning rate is used to back propagate the network and update the network weights.

Wang *et al.* proposed PPI specific Feature-Selective Rotation Forest (FSRF) algorithm for effective reduction in the data dimension and noise removal. Weighted values of

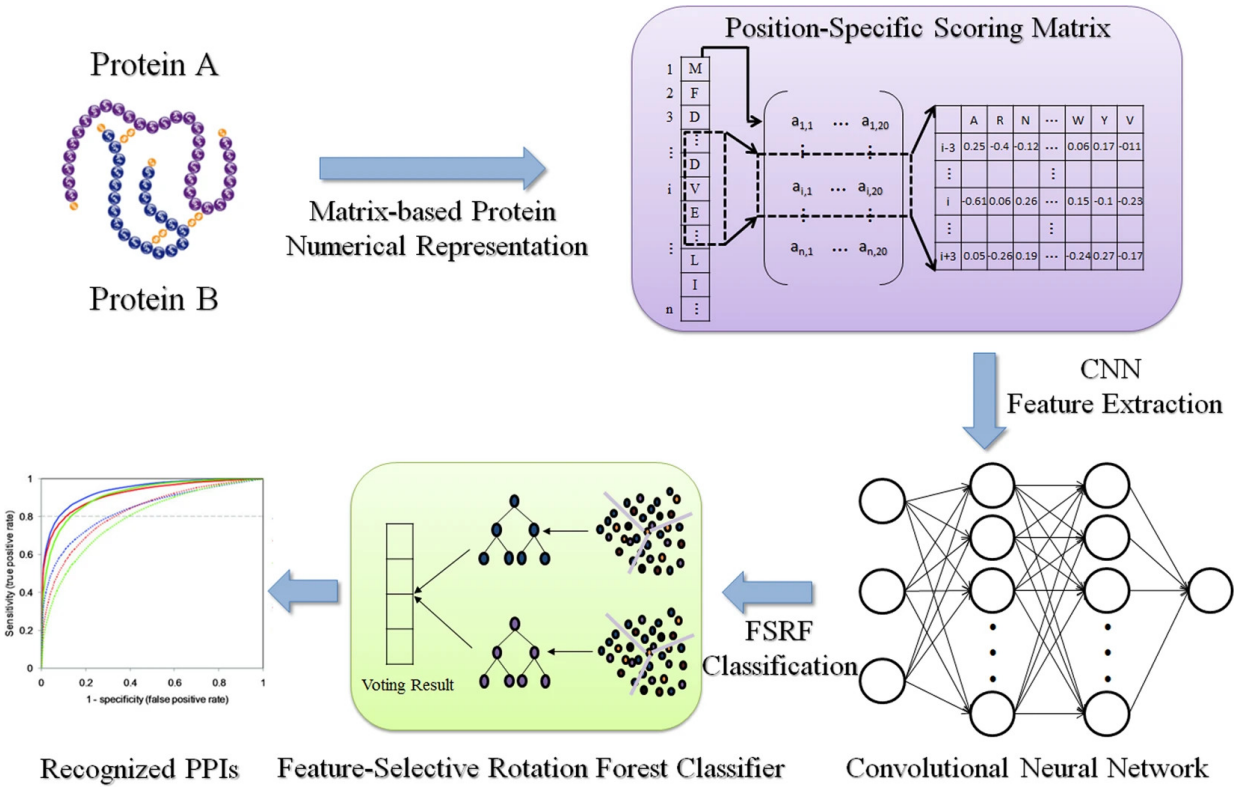

Figure 2.1: The schematic diagram for predicting PPIs by integrating CNN with feature-selective rotation forest model. Image courtesy : Wang *et al.* [1]

each feature was calculated using chi-square method and ranked to discard features having low influence on classification. A number of rotation matrices [23] along with the random forest classifier was used to finally assign the classes. The entire workflowvof the CNN-FSRF model has been depicted in figure 2.1 . They used 5-fold cross-validation and the widely used evaluation criteria to evaluate the model, including accuracy, sensitivity, specificity, precision, F-Score, and Matthews Correlation Coefficient (MCC). The experimental results show that CNN-FSRF performs significantly well in predicting PPIs.

2.3.3 LSTM-PHV: prediction of human-virus PPIs by LSTM with word2vec

A wide number of PPIs between human and virus are involved in viral infection. Sequence-based machine learning methods are expected to overcome the problems with expensive and time consuming experimental methods. In [24], Tsukiyama *et al.* first proposed the LSTM model with word2vec to predict PPIs between human and virus, named LSTM-PHV, by using only the amino acid sequences. The LSTM-PHV

effectively learnt the training data with a highly imbalanced ratio of positive to negative samples.

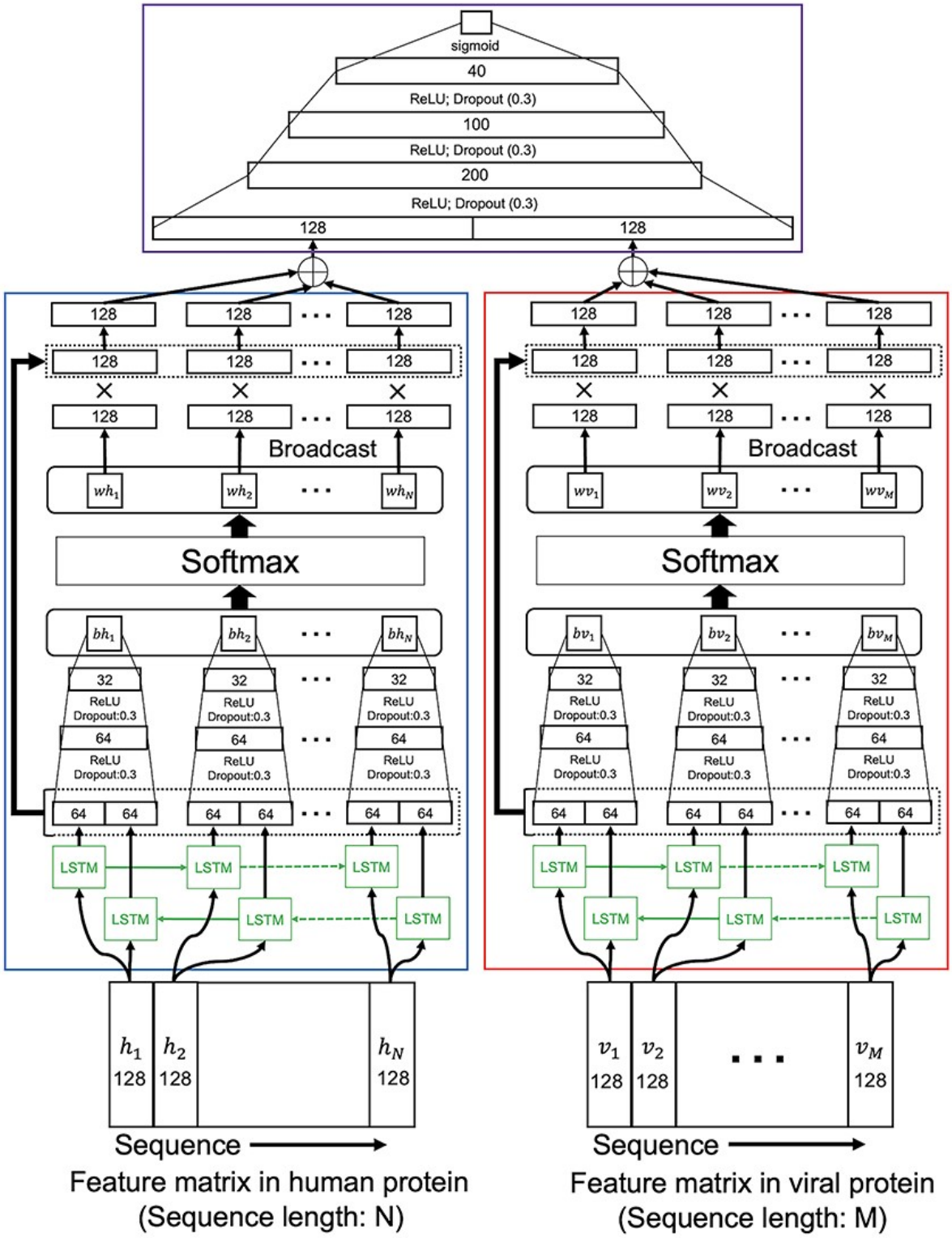

Figure 2.2: Network architecture of LSTM-PHV. Image courtesy : Tsukiyama *et al.* [25]

To address the problem of no gold standard data for negative samples they employed the dissimilarity-based negative sampling method which uses the Needleman–Wunsch algorithm of BLOSUM30 to determine the sequence similarities of all pairs of virus proteins in positive samples, and a similarity vector was defined for each virus protein followed by excluding the virus proteins showing lower sequence similarities. The weights in a neural network learn the context of words in word2vec using Continuous Bag-of-Words Model (CBOW), resulting in a distributed representation that encodes many linguistic regularities and patterns. Using the word2vec approach, the amino acid sequences of human and viral proteins registered as positive and negative samples were encoded as matrixes. In amino acid sequences, k-mers (k consecutive amino acids) were treated as a single word (unit), and each amino acid sequence was represented by numerous k-mers.

The LSTM-PHV is composed of three sub-networks. The human and virus proteins-embedding matrices were turned into two fixed-length vectors by two upstream networks with the identical structure. The PPIs were predicted using the third network's concatenated fixed-length vectors. Concatenated vectors are what they're called. Each step of the LSTM units receives the amino acid sequence column vectors from the embedding matrices. The rectified linear unit (ReLU) function with a dropout rate of 0.3 was employed as an activation function in the first two layers. The third layer's scalar values were aligned into a vector, which was then passed to the softmax function.

The resultant human and viral proteins' fixed-length vectors were concatenated in line and propagated into the final network. Three layers plus an output layer make up the final network. The output of the three layers was subjected to the ReLU function with a dropout rate of 0.3. The sigmoid function was employed as an activation function at the output layer to create a final output with a value between 0 and 1. They weighted a binary cross-entropy loss function in the way reported by Cui *et al.* [26] to train the model on unbalanced data using rectified adam (RAdam) optimizer [27].

Word2vec is capable of preserving information about local amino acid residue patterns. The architecture has been shown in figure 2.2. In comparison to previous state-of-the-art models, the LSTM-PHV learned highly unbalanced data and was able to reliably predict the interaction of a human protein with an unknown viral protein.

2.3.4 Structure-aware PPI site prediction using deep graph convolutional network

Graph convolutional network (GCN) [28] perform similar actions as CNN, in which the model learns the features by analysing surrounding nodes. The main distinction between CNNs and Graph Neural Networks (GNNs) is that CNNs are designed to work with normal (Euclidean) organised data, whereas GNNs are a generalised variant of CNNs in which the number of nodes connections varies and the nodes are not in any particular order (irregular on non-Euclidean structured data). GCNs and its variants have been successfully applied to a wide range of tasks with graph-structured data in recent years, including genomic analysis, protein solubility prediction, and drug discovery. Despite their success, most GCN-based models use shallow architectures that do not allow them to collect information from high-order neighbours. Yuan *et al.* in a recent article [29] proposed the GraphPPIS (deep Graph convolutional network for Protein-Protein Interacting Site prediction), deep graph-based framework for PPI site prediction, in which the PPI site prediction problem is transformed into a graph node classification task and solved by deep learning using initial residual and identity mapping techniques.

Residues that have been conserved throughout evolution may contain patterns related to crucial protein features like protein binding propensity. Therefore, to train the model, they used two types of amino acid features: position-specific scoring matrix (PSSM) and Hidden Markov models (HMM) as evolutionary information, as well as structural properties (DSSP), which were combined to generate the final node feature matrix.

Given a protein, they represented the protein graph by node feature matrix and the adjacency matrix. The edges in a protein graph were represented by an adjacency matrix, which was produced in two steps: 1) obtaining the coordinates of the Cα atom of each amino acid residue from the PDB file of a protein, and then calculating the Euclidean distances between all residue pairings, resulting in a distance map. 2) converting any value less than or equal to the chosen cutoff to 1 and any value larger than the cutoff to 0 to create an adjacency matrix from this protein distance map. Based on the model's performance on the training data, the cutoff was chosen at 14 Å. The graph convolutional operation is computed using ReLU activation function. Figure 2.3 represents the overview of how the model predicts it's final output class.

The multilayer perceptron uses the output of the last graph convolutional layer to estimate the protein-interacting probabilities of all amino acid residues. Finally, soft-

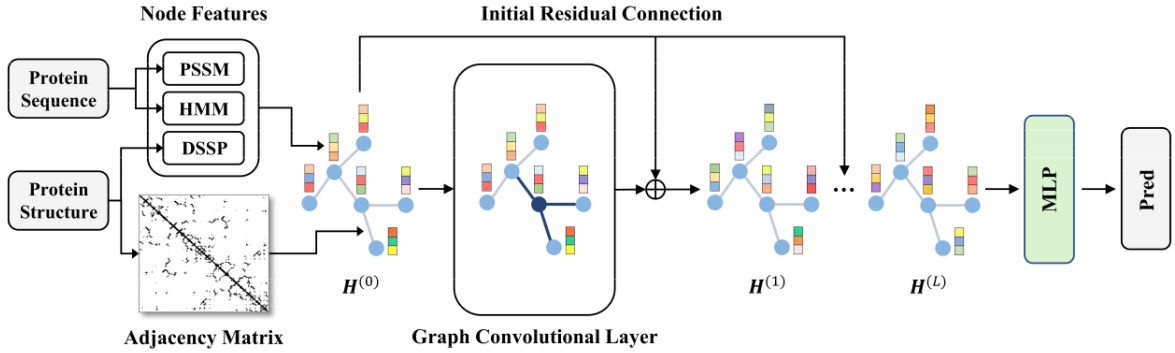

Figure 2.3: The network architecture of the proposed GraphPPIS model. The node feature matrix and the adjacency matrix are used as input into an L-layer graph convolutional network with initial residual and identity mapping. Here, H denotes the hidden state of the network and L is set to 8 in this work. The output of the L^{th} layer $(H^{(L)})$ is converted to residue-level protein-interacting probabilities by the final MLP module. Image courtesy : Yuan *et al.* [29]

max function transforms the network's output into a probability distribution for the two anticipated classes (non-interacting and interacting). This work is the first to utilize deep graph convolutional network for PPI site prediction, which can be easily extended to structure-based prediction of other functional sites. In comprehensive evaluations, GraphPPIS performs better than existing sequence-based and structure-based approaches. However, because the AUROC on the test set is less than 0.8, there is still space for improvement on this task.

2.3.5 Multi-level Feature Based Method for PPI Prediction with a Refined Strategy for Performance Assessment

ML-based models are frequently used in computational approaches to predict interactions based on high-throughput experimentally validated positive and negative interactions. Although experimentally confirmed protein-protein non-interactions (PPNI) databases are not generally available, high quality negative data are as important for classification and validation. Several deep learning-based techniques used in sequence-based PPI prediction have shown excellent results as a result of recent improvements. Regrettably, the majority of these techniques have a poor cross validation (CV) strategy and have completely overlooked the component level overlapping issue. When they are used on complex datasets, the performance suffers as a result of this flaw.

Generally, PPNIs are defined as a group of protein pairs that have never been reported to interact before. In particular, when modelling PPIs, we can think of the interaction

space as a graph $G(v, e)$, where v denotes a set of proteins and e is the set of known positive interactions between protein pairs. The set of edges (e') of the complement graph $G'(v, e')$ whose interactions are unknown can thus be described as PPNIs. Several deep learning-based techniques used in sequence-based PPI prediction have shown good results as a result of recent improvements. Unfortunately, the cross validation (CV) strategy has been inadequately handled in most of these solutions, and the overlapping issue in residue level has been completely overlooked. When applied to complicated datasets, the performance adversely affected as a result of this issue. Some of them are careless with their CV layout, while others are unconcerned with how positive and negative samples are chosen.

JUPPI [30], proposed by Halder *et al.* overcomes the above issue by introducing three level filtering to ensure high quality negative data with a pair wise CV method based on difficulty of the obtained data. The three level features include sequence, function (GO), and domain information. From the encoded sequence representation, they employed a trigram technique to build innovative min-max-sum based features. Amino acids has been divided into seven separate classes for sequence-based characteristics. They generated a trigram-based feature based on the encoded sequence, taking into account the local availability of encoded amino acids. There are $7^3(343)$ trigrams in the encoded sequence of seven unique symbols. In particular they calculated the measure of coordination (MCR) for the sub sequence pair of trigrams which occurs together when two proteins interact. For any protein pair P_a and P_b, the MCR score for k^{th} trigram α_k^{ab}has been calculated as :

$$MCR(\alpha_k^{ab}) = \frac{\eta_{min}(\alpha_k^{ab})}{\eta_{max}(\alpha_k^{ab})} log \frac{\eta_{min}(\alpha_k^{ab})}{\eta_{sum}(\alpha_k^{ab})} \tag{2.7}$$

where η_{min}, η_{max} and η_{sum} denotes the smaller, larger, and sum of local availability of k^{th} trigram pairs. The local availability for trigram pattern α_k^a is defined as :

$$\eta(\alpha_k^a) = \frac{\text{the number of occurrence of trigram } \alpha_k^a \text{ in } P^a}{\text{the sequence length of protein } P^a - 2} \tag{2.8}$$

Finally, the sequence based feature $f_{seq}(\alpha_k^{ab})$ for any protein pair P_a and P_b is calculated as :

$$f_{seq}(\alpha_k^{ab}) = \begin{cases} MCR(\alpha_k^{ab}), & \text{if } \alpha_k^a \times \alpha_k^b \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

The Gene Ontology (GO) graph is organized into three discrete directed acyclic graphs (DAG) based on three distinct protein features: cellular component (CC), molecular function (MF), and biological process (BP) . These graph representations have nodes

that correspond to certain GO-terms and edges that express various hierarchical relationships between them. The authors were able to derive six features by looking at more generic (ancestor) and more specific (descendant) level information of each GO-term pair from each of three types of GO relationship sub-graphs by normalising a ratio of immediate common ancestors/descendents to the set of immediate ancestors/descendents. Thus, in a representative GO sub-graph of type T, the ancestor specific $(\zeta_A^{T(g)})$ and descendent specific $(\zeta_D^{T(g)})$ directions for a pair of GO terms i and j is defined as :

$$\zeta_A^{T(g)}(i,j) = \left\{ \frac{|x|}{|y|} : x \in CA(i,j) \text{ and } y \in (A(i) \cup A(j)) \right\} \tag{2.10}$$

$$\zeta_D^{T(g)}(i,j) = \left\{ \frac{|x|}{|y|} : x \in CD(i,j) \text{ and } y \in (D(i) \cup D(j)) \right\} \tag{2.11}$$

where the collection of direct ancestors and descendants of each GO-term pair from each of three types of GO relationship sub-graphs is represented by $A(i)$and $D(i)$. $CA(i,j)$ and $CD(i,j)$ indicate GO-term i and j's immediate common ancestors and descendants, respectively. The GO-based characteristic is divided into two levels: the first is the GO-pair level, and the second is the protein-pair level. If there is a valid path from i to k on the general or particular direction of GO sub-graph T, a GO-term k will be deemed an ancestor or descendant of GO-term i. Finally, ancestor level feature $(\zeta_A^{T(p)})$ and descendant level feature $(\zeta_D^{T(p)})$ for a protein pair P_a and P_b has been defined as :

$$\zeta_A^{T(p)}(P_a, P_b) = \frac{1}{h \times v} \sum_{\substack{1 \le a \le h \\ 1 \le b \le v}} \zeta_A^{T(g)}(P_a, P_b) \tag{2.12}$$

$$\zeta_D^{T(p)}(P_a, P_b) = \frac{1}{h \times v} \sum_{\substack{1 \le a \le h \\ 1 \le b \le v}} \zeta_D^{T(g)}(P_a, P_b) \tag{2.13}$$

where h and v are the number of GO annotations of type T in P_a and $P_b$respectively. A domain, also known as a motif, is a structural and functional unit of a protein. Firstly, all possible interactions between different domains observed in the high throughput positive protein data has been taken into consideration and a subset of negative interaction pairs has been filtered out (based on *Negatome 2.0*) to assign the occurrence score to the

final set after normalization. Normalization is computed as:

$$f_{norm}^{ij} = \frac{O^{ij} - O_{min}}{O_{max} - O_{min}} \tag{2.14}$$

where O^{ij} is the occurrences of domain pair (d^i, d^j) and O_{max} and O_{min} are the maximum and minimum occurrence score respectively. Finally, domain affinity for a pair of protein P_a and P_b is defined by :

$$D_{affinity} = \sum f_{norm}^{d^a, d^b} \tag{2.15}$$

where $d^a \in dom(P_a)$ and $d^b \in dom(P_b)$ and $dom(P_i)$ is the set of domain annotations for P_i. To produce the final dataset, all three feature vectors were concatenated to produce a 350 (343 structural + 6 GO + 1 domain) dimensional feature vector for each pair of proteins.

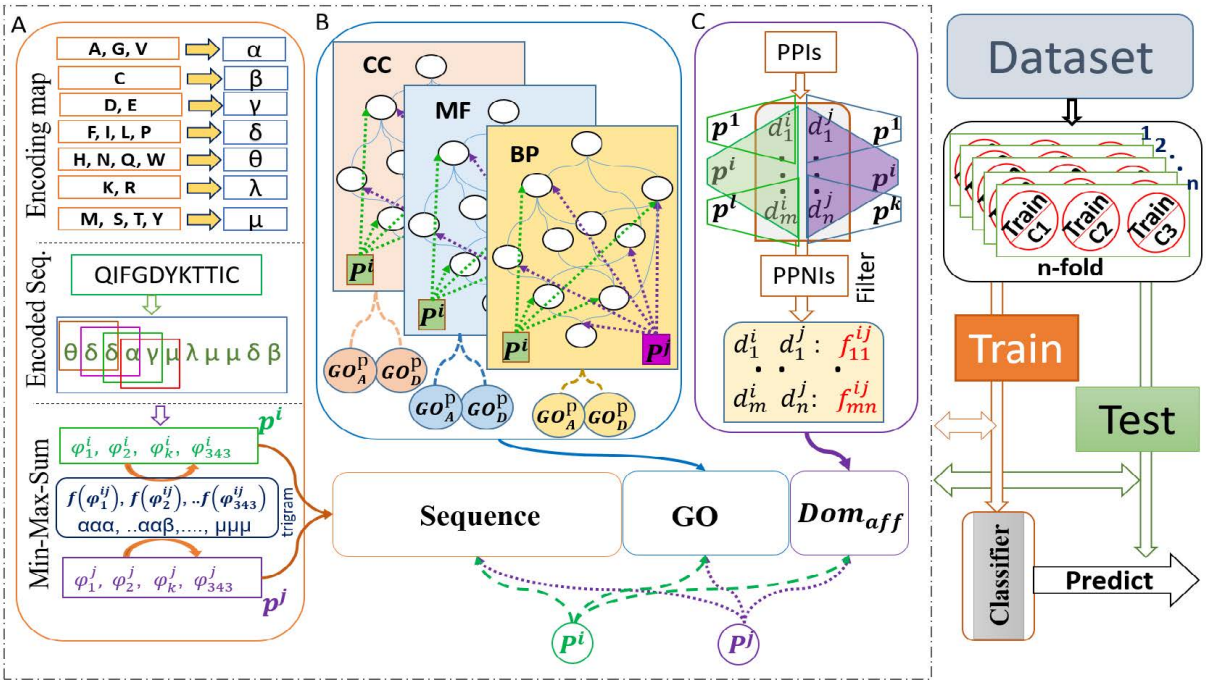

Figure 2.4: Basic workflow of JUPPI, A) Sequence Encoding, B) GO features:, C) Domain affinity. Image courtesy : Halder *et al.* [30]

The reliability of an interaction is depicted by HIPPIE's [31] interaction confidence scoring in the range of [0,1]. The final data is chosen based on PPIs with scores greater than 0.75. For data generation, the individual protein sequences of these interactions are collected as primary sequence bins. CD-Hit [17] with less than 40% identity is used to remove homologous sequences from the original sequence bins, ensuring null occur-

rences of redundant and homologous proteins between the train and test sets, as well as within the protein pair (interacting or non-interacting). However 3gClust [32] may also be used to remove redundant protein pairs.

The following three criteria have been used to curate negative data: 1) protein pairs with no interaction evidence in the positive set, 2) protein pairs from different subcellular locations where location is confirmed by manual assertion with experimental evidence and not implicated in any other location by any assertion method, and 3) protein pairs with no interaction evidence up to 3-level neighbours in the positive interaction network. Additionally, they developed the dataset or PPI prediction on three classes of prediction difficulties C1, C2 and C3, proposed by [33] by evaluating protein pairs from two graphs representing two non-redundant train and test sets, where edges represent interactions (positive and negative) and nodes represent the collection of proteins engaged in interactions. On three classes of prediction difficulty, C1, C2, and C3, the dataset was created to find the best classifier for PPI prediction. Graph representation can be used to predict the complexity of test classes. Consider $G_{train}(rE, rV)$ and $G_{test}(sE, sV)$, which represent two non-redundant train and test sets, respectively, with edges indicating positive and negative interactions and nodes representing the set of proteins participating in interactions. Three complex test classes are defined for any two proteins P_aX and P_b, which are as follows :

C1: both $< (P_a \in rV) \wedge (P_b \in rV) >$ and $< (P_a \in sV) \wedge (P_b \in sV) >$ can occur but not as a pair (P_a, P_b) .
C2: if $(P_a \in rV) \wedge (P_b \in rV)$, then either $< (P_a \in sV) \wedge (P_b \notin sV) >$ or $< (P_b \in sV) \wedge (P_a \notin sV) >$ can occur,
C3: if $< (P_a \in rV) \wedge (P_b \in rV) >$ then $< (P_a \notin sV) \wedge (P_b \notin sV) >$.

Finally, the classification task was performed using a Random Forest classifier addressing the class imbalance problem and the performance was evaluated by adding negative interactions and generating class imbalanced test datasets. The flowchart of JUPPI has been depicted in figure 2.4 for better understanding. JUPPI beats state-of-the-art approaches in terms of multiple metrics, according to experimental results on six separate datasets with varied complex test classes of prediction environment.

2.3.6 Deep Neural Network and Extreme Gradient Boosting Based Hybrid Classifier for Improved Prediction of Protein-Protein Interaction

Mahapatra et al. in this article [34] proposed a novel hybrid approach combining deep neural network (DNN) and extreme gradient boosting classifier (XGB) is employed for predicting PPI. Combining three sequence-based features as inputs : amino acid composition (AAC), conjoint triad composition (CT), and local descriptor (LD) creates the hybrid classifier (DNN-XGB). The raw features that are fed to the XGB classifier are abstracted layer by layer by the DNN in order to extract the hidden information. Saccharomyces cerevisiae (core subset), Helicobacter pylori, Saccharomyces cerevisiae, and Human have intraspecies interactions datasets, and their respective 5-fold cross-validation accuracy rates were 98.35, 96.19, 97.37, and 99.74 percent. The accuracy of the interspecies interaction datasets of the human-Bacillus anthracis and human-Yersinia pestis datasets, respectively, is 98.50 and 97.25 percent. The DNN-XGB can be utilised to forecast interspecies interactions, according to the increased prediction accuracy results obtained on the independent test sets and network datasets.

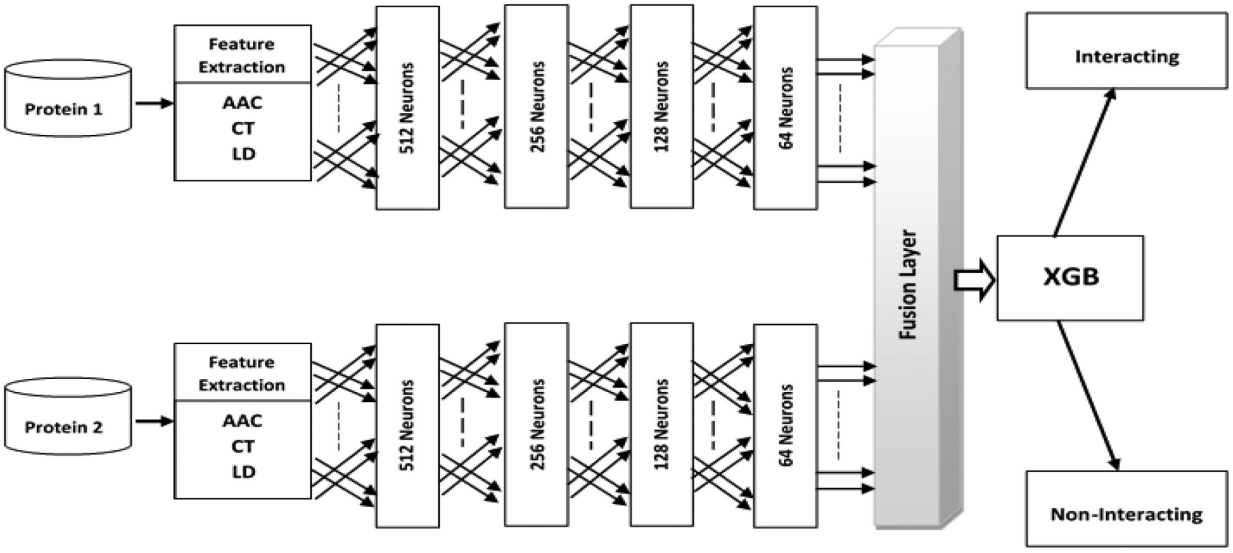

Figure 2.5: Schematic diagram of DNN-XGB classifier. Image Courtesy : Mahapatra *et al.*[34]

They concatenated the AAC, CT and LD features for both the proteins and used for feature extraction. In AAC, the frequency of each type of amino acid occurring in a protein sequence is computed and normalised with the length of the protein sequence. The CT and LD method as described as above. Thus finally creating a feature vector of shape 1×933 with a fusion of 20 AAC features, 343 CT features, and 630 LD features. Each

channel for a protein consisted of four fully connected layers of 512-256-128-64 neurons. Figure 2.5 represents a schematic diagram of their work. Finally, the output of these fused extracted features from two proteins are fed to extreme gradient boosting (XGB) classifier which an ensemble supervised learning algorithm based on gradient boosted trees. The main idea behind it is to make a strong classifier by combining predictions of weak classifiers following a serial training process. This is the current state-of-the-art model. We are trying to make a better model/architecture than this.

2.4 Comparison

To measure the prediction ability, the following metrics studies were carried out on different datasets. A comprehensive table for evaluation of performance of the methods discussed above have been prepared and is represented in Table : 2.2 to get an overall idea of the articles reviewed.

Table 2.2: Performance Evaluation for all the approaches that have been mentioned here.

Method	Dataset	Performance Evaluation						
		Sens.	Spec.	Prec.	Acc.	F-Score	AUC	MCC
Stacked AutoEncoder[5]	2010 HPRD NR	0.9806	0.9634	0.9627	0.9719	NA	NA	NA
	E. coli	0.9689	0.9528	0.9518	0.9605	NA	NA	NA
	C. elegans	0.9935	0.9528	0.9508	0.9723	NA	NA	NA
	Drosophila	0.9951	0.9628	0.9616	0.9784	NA	NA	NA
	DIP	NA	NA	NA	0.9377	NA	NA	NA
CNN-FSRF[1]	C. elegans	0.9641	NA	NA	0.9641	0.9817	NA	NA
	E. coli	0.9547	NA	NA	0.9547	0.9768	NA	NA
	H. sapiens	0.9865	NA	NA	0.9865	0.9932	NA	NA
	M. musculus	0.9327	NA	NA	0.9327	0.9652	NA	NA
LSTM-PHV[25]	TR1-TS1	0.906	0.829	NA	0.867	NA	0.912	0.737
	TR2-TS2	0.933	0.747	NA	0.84	NA	0.941	0.692
	TR3-TS1	0.892	0.822	NA	0.857	NA	0.921	0.716
	TR4-TS2	0.913	0.887	NA	0.9	NA	0.956	0.8
GraphPPIS[29]	Test_60	NA	NA	0.368	0.776	0.451	0.786	0.333
JUPPI[30]	Park-Marcotte : Human Balanced	NA	NA	NA	NA	NA	0.85	NA
	Park-Marcotte : Human Random	NA	NA	NA	NA	NA	0.87	NA
	Ramp-Host nr-human (CV)	NA	NA	NA	NA	NA	0.968	NA
	Yu et al. , Human : HCP:RNeg	NA	NA	NA	NA	NA	0.85	NA
DNN-XGB[34]	S.Cerevisiae	NA	NA	NA	0.9835	NA	NA	NA

Note: NA means Not Available
Sens. means Sensitivity
Prec. means Precision
Acc. means Accuracy
AUC means Area under ROC Curve
MCC stands for Mathews correlation coefficient

From the performance comparision table, it is observed that DNN-XGB has outperformed other models in accuracy. However, other models such as CNN-FSRF has obtained great results on inter-species predictions.

CHAPTER 3

Proposed Methodology

This section describes the approaches that has been incorporated in our work to achieve the Protein Protein Interaction (PPI) prediction task efficiently. Multiple amino acids (AAs) are joined together by peptide bonds to create a lengthy chain within a protein. A biological process removes a water molecule as it connects the amino group of one AA to the carboxyl group of a nearby AA to generate peptide bonds. The primary structure of a protein is defined as the linear sequence of amino acids inside it. The raw AA sequence from a pair of proteins has been converted into an image and fed to DenseNet [35] Deep Learning (DL) framework to generate two models to focus on class imbalance problem. Finally, the classification result has been obtained by performing a strict threshold on the results derived from the generated models. DenseNet framework has been used here as it has most accurate representation of images, improved parameter adaptability, higher memory and computational efficiency. DenseNet is a novel Convolutional Neural Network (CNN) architecture that has achieved *state-of-the-art* results on various classification datasets with less parameters than previous image classifiers like Convolutional Neural Network (CNN) and Residual Networks (ResNet), hence it is highly likely to outperform them.

3.1 Dataset

Several standard intraspecies (Human) datasets are utilised in this study. Pan's PPI data has been obtained from [36] as a benchmark dataset to train our models with positive and negative PPIs. There were 36,630 positive and 36,480 negative pairs from 9476 and 2184 proteins in the sample respectively. The remaining 36,630 interacting (positive) samples (PPIs) were generated by eliminating duplicated interactions from the human protein references database (HPRD, 2007 edition). However, the non-interacting (negative) dataset has been constructed using pair of proteins, each of which are from different sub-cellular regions of the Human species. The sub-cellular region information was curated from Swiss-Prot version 57.3 database based on the sequence having ambiguous or uncertain subcellular location terms, two or more sites, with the annotation "fragment" and sequence length ≤ 50. Sequences with length more than 20,000 were discarded for being very computationally expensive resulting in 36,608 and 36,480 positive and negative interactions respectively. We randomly selected 4500 positive and 4500 negative interactions and used it as hold-out set for further validation. The remaining data has been used to create three models, first of which uses all of the remaining PPIs with all the sub-images (discussed in section 3.3 and 3.4) generated from the PPIs and second one uses approximately equal amount of sub-images (positive and negative) of Pan et al.'s dataset and the third model uses dataset of whole PPI dataset of *Saccharomyces cerevisiae* (S.Cerevisiae) to train the Deep Learning (DL) model, which contains 17257 interacting pairs and 48954 non-interacting pairs, was used to train a separate model on the same architecture in order to predict its generalisation potential on the mentioned inter-species datasets. In the PPI datasets, the number of samples in both the positive and negative classes is not equal, resulting in an unbalanced data set. As a result, a balanced dataset is created by selecting a number of negative samples equal to the number of positive samples thus creating three DensePPI models with names **DensePPI-PE** (DensePPI-Pan et al. "Equal"), **DensePPI-PF** (DensePPI-Pan et al. - "Full") and **DensePPI-SC** (DensePPI-S.Cerevisiae). For creating a low complexity consensus based DL based model, we randomly selected 4500 positive and 4500 negative interactions and held them out for further validation. The remaining data has been randomly chosen and equally divided into 10 folds for performing the 10 fold CV where each fold had 3210 and 3198 positive and negative exclusive dataset. We named this model as **DensePPI-CN** (DensePPI-Consensus).

3.1.1 External human validation databases

In order to verify the prediction and generalization capability of our model, we used several external databases to obtain and compare our model's performance. The first database that we chose to use was of DIP. The Database of Interacting Proteins (DIP) is a record of experimentally determined PPIs [37]. Specifically, 20160430 version of DIP for Human species has been used for validating our models. The second database, Human Integrated Protein–Protein Interaction rEference (HIPPIE) provides annotated PPIs and their confidence scores using graph algorithms [38]. The PPIs with confidence score $\geq$ 0.73 has been considered as High-Quality (HQ) data and the PPIs with score $<$ 0.73 has been considered as Low-Quality (LQ) datasets. And lastly, InWeb_InBioMap(IWIBM) PPI database, which is a combination of 8 huge PPI databases to provide a scored human protein interaction network through data integration and quality control, with greater number of interactions and higher functional biological significance than comparable resources [39]. This data discriminated between two forms of PPI data: HQ data, which had a confidence value of 1, and used rest of the interactions as LQ data. The recent release of IWIBM was utilised to compare our results. The DIP, HIPPIE and IWIBM databases contained 6913, 287357 and 625641 amount of positive interacting PPIs in total.

3.1.2 External independent validation databases

To further study the prediction capability of our model, we tested our model on the independent databases of different species obtained from [34]. We collected *Caenorhabditis elegans* (C.Elegans), *Escherichia coli* (E.coli), *Helicobacter pylori* (H.Pylori), *Homo sapiens* (H.Sapi), *Mus musculus* (M.Musc) data contained 4013, 6954, 1420, 1412 and 313 positive interactions respectively.

All the datasets obtained were further pruned of protein pairs that were similar to the benchmark dataset. We used CD-Hit [17] program to identify and remove protein pairs whose sequences were 40% similar to benchmark dataset as it can handle large datasets and is hundreds of times faster than common database search and sequence comparison tools.

3.2 Image Generation from Sequences

Sequence from both the proteins in a PPI pair has been used to generate an image to be used for training and final classification. As protein is a long chain of texts (AA sequence), we have considered equal connection strength for interaction between two proteins where connections are of number N×M for two proteins with N and M number of AAs respectively. The 20 AAs 'M', 'D', 'A', 'K', 'R', 'G', 'L', 'C', 'V', 'F', 'S', 'P', 'Q', 'E', 'I', 'H', 'Y', 'T', 'W', 'N' along with other special cases which are hard to differentiate such as 'O', 'U', 'B', 'Z', 'X', 'J' has been assigned the most distinct 26 colours in RGB colour spectrum chosen using [40] in order to provide equal importance for each AA interaction. We then prepared a colour map (CMAP) which is a colour matrix of dimension 26×26 where each entry CMAP_{ij} represents a particular colour for the interaction of residues i and j is defined as :

$$\text{CMAP}_{\text{ij}} = \left[\sqrt{\frac{r_{\text{i}}^2 + r_{\text{j}}^2}{2}}, \sqrt{\frac{g_{\text{i}}^2 + g_{\text{j}}^2}{2}}, \sqrt{\frac{b_{\text{i}}^2 + b_{\text{j}}^2}{2}} \right] \tag{3.1}$$

where $[r_{\text{i}}, g_{\text{i}}, b_{\text{i}}]$ and $[r_{\text{j}}, g_{\text{j}}, b_{\text{j}}]$ are the unique colours for residues i and j where $r, g,$ and b are the colour intensity values of red, green, and blue primary colours in the RGB colour model within the range [0, 255]. These colour maps for a particular interaction between two AAs i and j i.e. CMAP_{ij} was further used to create colour matrix for interactions between AA sequences of two proteins P_1 and P_2. The final image (colour matrix) PCMAP of the two proteins is defined as:

$$\text{PCMAP}_{\text{ij}} = \text{CMAP}_{\text{ij}}; \forall \text{i} \in \text{AA}[\text{P}_1] \text{ and } \forall \text{j} \in \text{AA}[\text{P}_2] \tag{3.2}$$

where $\text{AA}[\text{P}_1]$ and $\text{AA}[\text{P}_2]$ represents the entire AA sequence of two proteins P_1 and P_2. Thus, images were generated and saved for all possible protein pairs to be trained and validated on our proposed deep learning architecture.

3.3 Sub-Image Generation

The images thus generated from protein pairs were further split into sub-images using sliding window approach. We used this method to generate equal size sub-images from

an image rather than squeezing the image as it would have lost information for images produced from proteins with larger sequences. The sub-image generated would have equal resolution and aspect ratio of each region as there were in the original image assuming equal prediction capability for all the sub-images generated from an image. A sliding window of a fixed dimension was hovered on the entire image with fixed stride which controls the amount of movement in horizontal and vertical direction over the image. All of the sub-images that resulted from a positive protein interaction were labeled as positive interactions and resulting sub-images from a negative protein interaction was labeled as negative interactions. The list of all generated sub-images from a particular image has been kept in track for further experimentation. Benchmark and external validation data has been used for generating sub-images with chosen sub-image sizes of 256×256 and 128×128 with a stride of 128 and 64 respectively to generate images of equal dimension without resizing it for the deep learning architecture to be discussed in next section. Generating sub-images from an image helps us to assign equal weight-age to each interaction regardless of size of both the proteins.

3.4 Creating datasets with equal number of sub-images

To create a dataset with approximately equal amount of sub-images from the remaining amount of images in training dataset, we used a local search on the dictionary mapping of image to it's number of sub images. In particular, we searched for the number of allowable sub-images for an image of a PPI for which we needed a threshold t_{img} which minimizes the following function :

$$|\sum \text{SI}_p - \sum \text{SI}_n| \tag{3.3}$$

where SI_p's are the total number of positive sub-images using only those positive images whose number of sub-images are $\leq t_{\text{img}}$ and SI_n's are the total number of negative sub-images using the same threshold t_{img}. Thus, we generated two models with Pan et al.'s "full" and "equal" dataset representing equal number of PPI images and approximately equal number of PPI sub-images respectively.

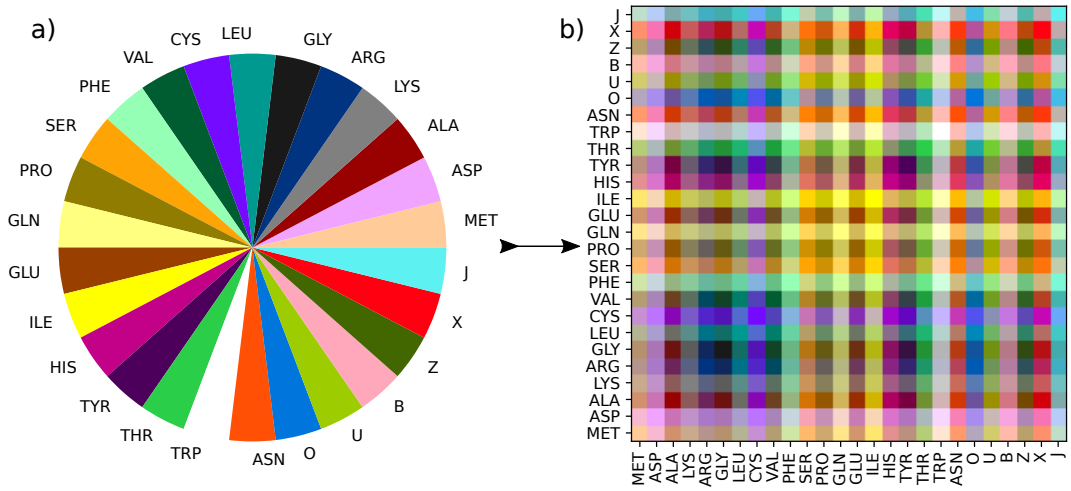

Figure 3.1: Assigned colours and colour maps to produce images from AAs in PPIs. a) Colour assigned to each amino acid and the unrecognizable amino acids. b) The colour map used for generating images from two proteins using amino acid pairs

3.5 Deep Neural Network Model

For visual object detection, convolutional neural networks (CNNs) have become the dominant machine learning approach. The original LeNet5 [41] has five layers, VGG has 19 layers [42], whereas Highway Networks [43] and Residual Networks (ResNets) [44] just recently breached the 100-layer barrier. As CNNs get more complicated, a new research concern arises: information about the input or gradient might “wash away" as it passes through several layers before reaching the network’s finish (or beginning). The vanishing-gradient problem was solved, feature propagation was improved, feature reuse was promoted, and the number of parameters was cut in half using Huang *et al.*’s Dense Convolutional Network (DenseNet).

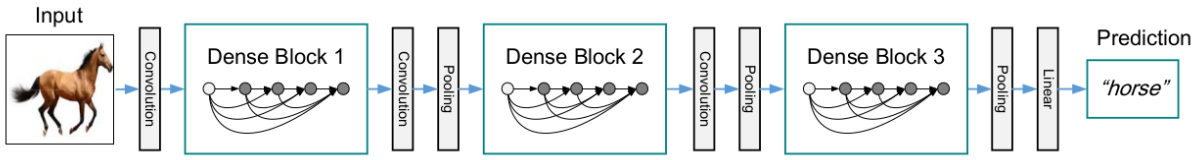

Figure 3.2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks (transition layers) change feature-map sizes via convolution and pooling. Image courtesy : Huang *et al.*[35]

Each layer in DenseNet is linked to every other layer in a feed-forward manner. It’s network has $\frac{L(L+1)}{2}$ direct connections, whereas normal L-layer convolutional networks have L connections, i.e. one between each layer and the next layer. Each layer uses the feature-maps of all previous levels as inputs, and its own feature-maps are utilised as in-

puts into all subsequent layers, i.e. an i^{th} layer receives feature-maps from it's preceding layers, $L_0, L_1, \ldots, L_{i-1}$ as an input:

$$L_i = f_i([L_0, L_1, \ldots, L_{i-1}]) \tag{3.4}$$

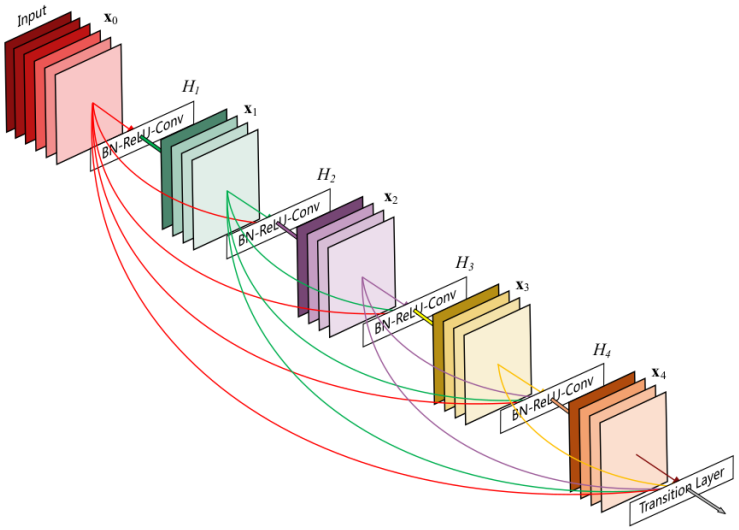

Figure 3.3: A 5-layer dense block with a growth rate of $k = 4$. All preceding feature-maps has been used as an input to each layer. Image courtesy : Huang *et al.*[35]

where L_i is the output of layer i and $[L_0, L_1, \ldots, L_{i-1}]$ is a concatenation of the feature-maps of previous layers. They defined f_i as a composite function of three successive operations: batch normalisation (BN), rectified linear unit (ReLU) and a 3×3 convolution (Conv). As a result, the feature-maps created by earlier layers may be easily accessible by the network deep levels, allowing the features to be reused. The network may be smaller and more compact since each layer gets feature maps from all preceding layers, resulting in fewer channels. Therefore, it is more efficient in terms of processing and memory. Their architecture divides the network into numerous tightly linked *dense blocks* to make downsampling easier. *Transition layers*, which perform convolution and pooling, are referred as layers between blocks. However, as layers go deeper, the number of concatenated feature-maps of the next layer increases. If there are no limitations set on the continued rise in the number of feature-maps, there might be a massive computing expense. Therefore, by adjusting the growth rate k, the amount of newly created feature-maps in each layer may be kept under control. When using a i number of layers, the i^{th} layer in the dense block will contain a total of $k \times (i-1) + k_0$ feature-maps, where k_0 denotes number of input channels into the dense layer. Figure 3.3 represents a 5-layer dense block and a DenseNet architecture has been shown in figure 3.2.

Table 3.1: Description of the DenseNet201 architecture

Layers	Output Size	DenseNet-201 Architecture
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Transition Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool
		1000D fully-connected, softmax

In particular, we used DenseNet201 implementation from using PyTorch [45] and keras [46], the architecture of which has been described in Table 3.1. The sub-images along with it's labels has been used as an input here to predict the classification results after discarding protein pairs whose at least one sequence length is < 128 so as not to squeeze or expand any pictures for generating sub-images from it. In addition to that, we chose *average* pooling between layers with a learning rate of 0.001 and an added momentum of 0.9. To train the pre-trained DenseNet model, we used all of the sub-images from the training dataset. Because the prediction task is stated as a binary classification problem, the *categorical crossentropy* measure was chosen as a loss function. The *categorical crossentropy* loss function is defined as :

$$J_{CCE} = -\frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N} y_j^i \times \log(h_\theta(x_j, i)) \tag{3.5}$$

where N and C denotes the number of training examples and classes respectively, x_j and y_j^i represents the j^{th} input vector along with it's target label for class i respectively and h_θ denotes the model with network weights θ. The networks are optimised using a stochastic gradient descent (SGD) optimizer to perforachieve generalization using Py-

Torch and keras to get the highest accuracy.

These tests are conducted on a workstation that has two NVIDIA Quadro P5000 GPUs with 16 GB of VRAM each, Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz, and 512 GB of RAM. A 64-bit version of Ubuntu 16.04 runs on the computer.

3.6 Classification Strategy

We adopted two classification strategies one of which uses consensus and other one uses whole data of PPI and PPNI dataset.

3.6.1 Strategy A : Consensus based classification

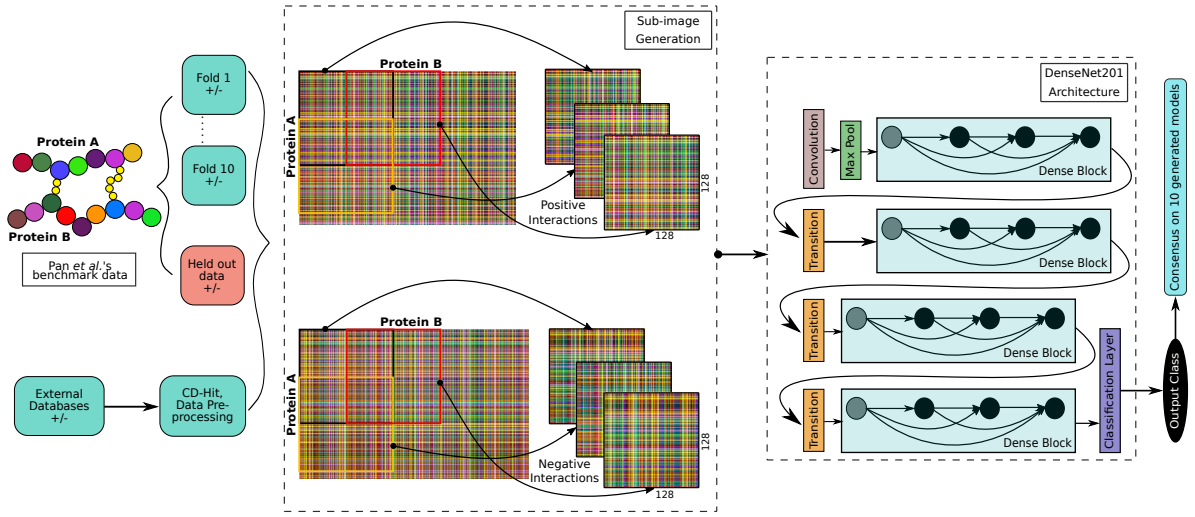

Figure 3.4: Overall work flow diagram of the methodology 1 proposed in this thesis. The last 4 blocks represent Dense blocks and other transition layers in the DenseNet201 architecture.

The confidence values $\in [0, 1]$ for each sub-image generated from an original image of a PPI obtained using the DenseNet201 prediction has been averaged out to get an near approximate overall confidence value of the DL architecture on the particular PPI. As we have chosen to perform 10-fold CV, we have obtained 10 models using equal amounts of positive and negative protein interaction data. The models were further subjected to an iterative method to find out the best threshold giving out the maximum score in the performance evaluation metrics using the test data belonging to the particular fold. We used accuracy metrics as a criteria to identify the best threshold for a particular model.

The PPIs under that threshold are considered as a negative interaction (0) and those which are $\geq$ threshold are considered as a positive interaction (1).

After receiving the best thresholds for a model, We then fed the held out data into 10 generated models and use the obtained thresholds to get 10 final classification result on each PPI. However, The final classification report for one PPI has been generated using consensus which is formulated below :

$$\mathrm{FC_i} = \begin{cases} 1, & \text{if } \mathrm{count}_i(1) \geq \mathrm{count}_i(0) \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

In the above equation, $\mathrm{FC_i}$ denotes the final class for the i^{th} PPI and $\mathrm{count}_i(1)$ and $\mathrm{count}_i(0)$ represents the number of times where a model has predicted 1 and 0 for the i^{th} PPI respectively.

3.6.2 Strategy B : Whole data based clustering

The confidence values $\in [0, 1]$ for each sub-image generated from an original image of a PPI obtained using the DenseNet201 model prediction has been averaged out to get an near approximate overall confidence value of the DL architecture on the particular PPI. We have obtained two models using equal number of positive and negative images and another with equal amounts of positive and negative sub-images. A threshold of 0.5 has

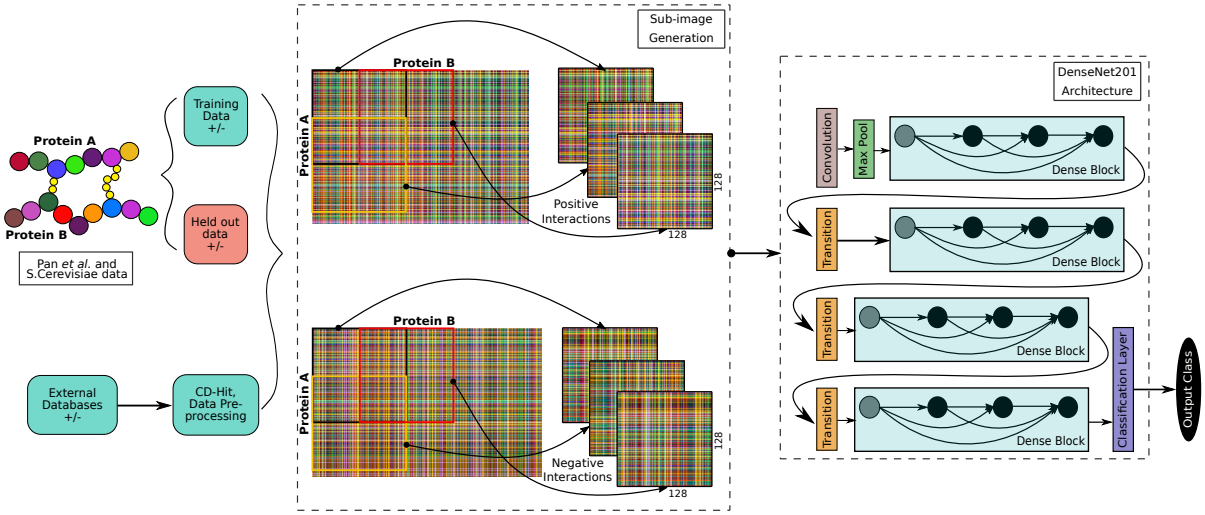

Figure 3.5: Overall work flow diagram of the methodology 2 proposed in this thesis. The last 4 blocks represent Dense blocks and other transition layers in the DenseNet201 architecture.

been chosen to give the final class label to the original PPI. Thus, the final classification report for one PPI has been calculated as :

$$\text{FC}_\text{i} = \begin{cases} 1, & \text{if avg}_i(\text{P}) \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

In the above equation, FC_i denotes the final class for the i^{th} PPI and $\text{avg}_i(\text{P})$ is the mean of all predicted probabilities for the sub-images of i^{th} PPI.

3.7 Performance Evaluation

The classification Accuracy, Area under Receiver Operating Curve (AUROC), Area under Precision Recall Curve (AUPRC), and the F1-score are used in this research article to evaluate the prediction models using the 10-fold cross-validation test. AUROC and AUPRC are based on Sensitivity, Specificity and Precision which are defined by :

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{3.8}$$

$$\text{Precision / Positive Predictive Power} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.9}$$

$$\text{Recall / Sensitivity / True Positive Rate} = \frac{\text{TP}}{\text{TP+FN}} \tag{3.10}$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{TN+FP}} = \text{1-Specificity} \tag{3.11}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{3.12}$$

The AUROC is a probability curve that plots the True Positive Rate against False Positive Rate, and is a metric for a classifier's ability to discriminate across classes. On the other hand AUPRC curve plots between Precision and Recall and mainly focusses on true positives. The AUROC, AUPRC curves has been reported as a score using sklearn Python package [47] to calculate performance metrics for each label, and find their unweighted mean.

The entire workflow of our proposed approach along with the DenseNet201 architecture has been depicted in figure 3.4 and 3.5.

CHAPTER 4

Results

This chapter describes the key findings and observations of the classification task so performed on the previous chapter. Detailed result obtained from all the analysis has been discussed here.

4.1 Data preparation

The raw Amino Acid (AA) sequence of the benchmark as well as the external validation dataset has been converted into image by using the methodologies described in section 3.2. The colour profiles for the amino acid pairs were assigned randomly from the set of 26 colours mentioned in [40]. The colour map of interaction between two proteins when the interact is shown in figure 3.1-(b). Colour assigned for each AA and the other unrecognizable AAs is depicted in figure 3.1-(a).

The benchmark datasets of Pan et al. and S.Cerevisiae was split into training and a held out portion to test the results obtained so far as discussed in section 3.1. We chose to use 8:2 ratio between number of training and testing images for Pan et al.'s PPIs. For the S.Cerevisiae benchmark dataset, 9:1 ratio between training and testing samples has been used for having low amount of PPI data. Although, not all the samples had both

dimensions ≥ 128, therefore some samples were discarded and thus three models with names **DensePPI-PE** (DensePPI-Pan et al. "Equal"), **DensePPI-PF** (DensePPI-Pan et al. - "Full") were generated and **DensePPI-SC** (DensePPI-S.Cerevisiae) and used these models to predict on the inter-species as well intra-species PPIs.

The entire benchmark dataset of Pan et al. was split into 10 equal folds along with a held out portion to test the results obtained so far as discussed in section 3.1. Thus, each fold has got 2568, 2558 number of positive and negative training samples with 642, 640 number of positive and negative testing samples respectively. The standard 8:2 ratio between number of training and testing items was maintained. Although, not all the samples had both dimensions ≥ 128 in each fold, therefore some samples were discarded. Table 4.1 shows us the percentage of images selected for training/testing for all folds for the **DensePPI-CN** (DensePPI-Consensus) model.

Table 4.1: Data distribution in each fold in CV for DensePPI-CN model

Model No.	No. of training Sub-images	No. of testing Sub-images	Percentage of image selected
1	421674	101051	91.533
2	417092	102600	91.475
3	416916	101305	91.650
4	407640	98435	91.280
5	422917	108092	91.455
6	445774	111389	91.046
7	416951	109079	91.592
8	425287	99249	91.377
9	425026	102260	91.280
10	415484	98754	92.041

4.2 Sub-image Generation

We tried squeezing the images in order to reduce time complexity of handling huge number of sub-images. But the accuracy obtained were much lower. An accuracy of 75.85% was obtained on squeezing the image to size 256×256, however when tried with our approach mentioned in 3.3, an accuracy of 86.038% was obtained. Original PPIs converted to image were further sub-divided using a sliding window of size 256×256 and 128×128 with strides of 128 and 64 respectively to compare the settings for generating sub-images giving maximum performance. It was found that sub-images generated using 128×128 window size with strides of 64 yielded better accuracy than the former combination. Consequently, sub-images of size 128×128 with a stride of 64 generating

input images of dimension 128×128×3 for the DenseNet model with 10 epochs was chosen as the best setting for DensePPI-PF, PE and SC models and was used as a parameter while running each model. Additionally, we also found the t_{img} as 154. Therefore, the number of positive and negative images with total sub-images for the original PPI images of all the models has been reported in table 4.2. Table 4.1 reports the number of sub-images from original training and testing images in each fold for the DensePPI-CN model.

Table 4.2: Sub-image and original image counts of the three models

Counts ↓ / Models →		DensePPI		
		PF	PE	SC
Orig Image Count	POS	30266	26598	15487
(Training)	NEG	32385	33588	15793
Sub Image Count	POS	3320234	1348570	1133820
(Training)	NEG	2207561	1346035	1026082
Orig Image Count	POS	4227	4227	1725
(Hold Out)	NEG	4003	4003	1725
Sub Image Count	POS	460952	460952	113206
(Hold Out)	NEG	280983	280983	103271

The whole dataset of interspecies and intraspecies PPIs has been used for redundancy removal with the particular benchmark PPI dataset. For Pan et al.'s data we used C.Elegans, E.Coli, H.Pylori interspecies and DIP, HIPPIE and IWIBM intraspecies PPI dataset for removal of sequence similarity using CD-Hit. Additionally C.Elegans, E.Coli, H.Pylori, H.Sapiens and M.Musculus interspecies PPIs has been used with the benchmark S.Cerevisiae data to reduce the redundancy upto 40%. The original image count and their reduced count based on sequence similarity with their sub-image count has been reported in table 4.3.

4.3 Results from DenseNet Architecture

The DenseNet DL model was implemented according to the parameters discussed in section 3.5. The Accuracy, Sensitivity, Precision, F-score, Mathews Correlation Coefficient(MCC) score, Area under Receiver Operating Curve(AUROC) and Area under Precision Recall Curve(AUPRC) of our models on hold out data and it's comparison with state-of-the-art methods has been shown in table 4.4.

Our DenseNet DL architecture performed surprisingly well for the hold out data from the benchmark dataset, the data distribution of which is described in section 3.1. We re-

Table 4.3: Image counts of all redundancy removed external datasets using CD-Hit

Benchmark Data	CD-Hit used on	Orig IMG Count	LR IMG Count	Sub-IMG Count
DensePPI-SC	C.Elegans	4013	3403	201546
	E.Coli	6954	5779	152216
	H.Pylori	1420	1349	49005
	H.Sapiens	1412	1104	86172
	M.Musculus	313	256	17254
DensePPI-PF	C.Elegans	4013	2405	149437
	E.Coli	6954	5610	147994
	H.Pylori	1420	1329	48618
	DIP	6913	933	47205
	HIPPIE-HQ	45316	2727	132120
	HIPPIE-LQ	242041	37627	1902546
	IWIBM-HQ	166033	32788	1074202
	IWIBM-LQ	459608	78353	4049978

Note: LR stands for Low Redundancy
HQ stands for High Quality
LQ stands for Low Quality

ceived the best AUROC, AUPRC, Accuracy(Acc), Sensitivity(Sens), Precision(Prec.), F-score and MCC score for the DensePPI-PF model. It has been noticed that our DensePPI-PF model has outperformed all the other standard models viz. DNN-PPI[16], SAE[5] in the comparative study reported in table 4.4.

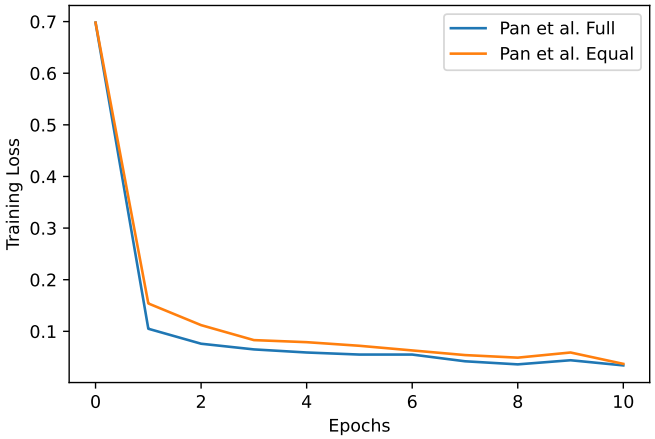

Figure 4.1: Training loss/ Loss convergence plot of DensePPI-PF and DensePPI-PE models for 10 epochs.

We plotted the loss comparison curves of the models in figure 4.1 to ensure that they were all convergent. In terms of the loss value, it was discovered that DensePPI-PF model had a faster convergence speed than the other two models with DensePPI-PE model converging at the nearest speed. Moreover, we also plotted the AUROC and

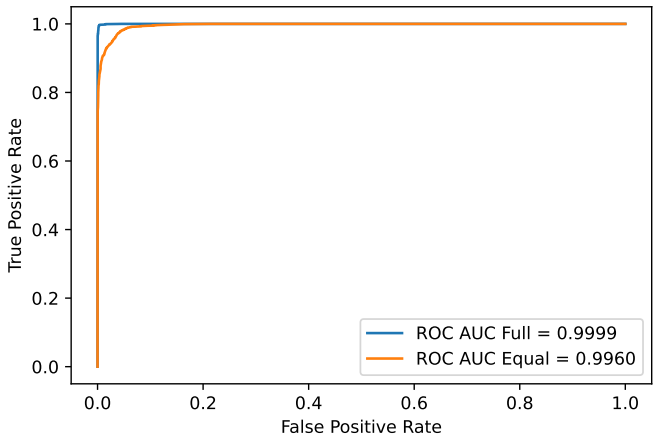

Figure 4.2: AUROC plot of DensePPI-PF and DensePPI-PE models with score.

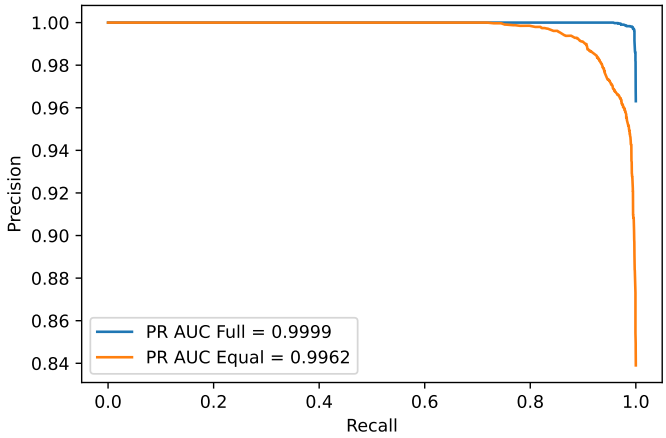

Figure 4.3: AUPRC plot of DensePPI-PF and DensePPI-PE models with score.

Table 4.4: Performance comparison of the models with DNN-PPI and SAE standard models

Models→	DNN-PPI[16]	SAE[5]	DensePPI		
			PF	PE	SC
Acc (%)	0.9878	0.9719	0.9957	0.9666	0.8958
Sens (%)	0.9891	0.9806	0.9946	0.9872	0.9513
Prec (%)	0.9861	0.9627	0.9972	0.9497	0.8548
F-Score (%)	0.9876	NA	0.9959	0.9681	0.9005
MCC (%)	0.9757	NA	0.9915	0.9338	0.7969
AUC (%)	NA	NA	0.9999	0.9960	0.9578
AUPRC (%)	NA	NA	0.9999	0.9962	0.9421

Note: NA means Not Available

S.Cerv stands for S.Cerevisiae model

AUPRC plot for the DensePPI-PE and DensePPI-PF model, which are reported in figures 4.2 and 4.3 respectively.

Table 4.5: Performance on training the 10 models along with the threshold giving best results for test data

Model No	Training Acc. (%age)	Threshold for max. performance	Testing Acc. (%age)
1	99.72	0.48	95.299
2	99.83	0.34	95.013
3	99.82	0.2	94.898
4	99.73	0.16	95.339
5	99.74	0.15	95.266
6	99.84	0.16	94.416
7	99.75	0.17	95.081
8	99.85	0.47	95.819
9	99.72	0.39	94.958
10	98.95	0.14	96.371

The DensePPI-CN DL model was implemented according to the parameters discussed in section 3.5. We obtained 10 thresholds for the 10 trained models in CV which provides maximum performance of the model when tested using the test samples of that fold. The training accuracy, testing accuracy and the threshold to be used for attaining best accuracy in test data has been reported in table 4.5.

Table 4.6: Performance of the 10 generated models along with the consensus approach results on the held out benchmark data

Model No	Result on Hold Out data			
	Accuracy (%age)	AUROC	AUPRC	F1-Score
1	94.605	0.972	0.968	0.93
2	95.383	0.98	0.978	0.901
3	94.994	0.975	0.972	0.872
4	95.176	0.975	0.974	0.875
5	95.310	0.972	0.972	0.866
6	94.800	0.978	0.976	0.876
7	95.237	0.974	0.97	0.888
8	95.128	0.978	0.977	0.927
9	93.742	0.969	0.968	0.931
10	94.727	0.973	0.969	0.884
Consensus	97.303	0.973	0.982	0.884

Our DensePPI-CN DL architecture performed surprisingly well for the hold out data from the benchmark dataset, the data distribution of which is described in section 3.1. We received the best Area under Receiver Operating Curve (AUROC) score of 0.98 for model number 2, best Area under Precision Recall Curve (AUPRC) score of 0.982 for the

consensus model and maximum F1-score of 0.931 for the 9^{th} DL model. The performance metrics thus obtained on all the models as well as while using the consensus approach is depicted in table 4.6.

4.4 Performance on external validation datasets

In the domain of machine learning, generalisation and overfitting are two significant and closely linked concepts. Overfitting refers to a model that fits training data too well, whereas generalisation refers to a model's capacity to adapt to new data. Despite of having manually curated dataset partitions and algorithm design perspectives, the problem persists due to the unpredictability of future unknown data. The most practical technique to assess a model's generalisation is still individual verification on each dataset.

We checked our DensePPI-CN model's performance on external human databases *viz.* DIP [37], HIPPIE v2.0 [38] and inWeb_inbiomap [39]. It has attained great performance on intraspecies databases also, the report of which is shown in table 4.7. We have compared our model's performance with standard models like SAE[5] and DNN-PPI[16].

Table 4.7: Intraspecies performance comparision of DensePPI-CN model

Models	DIP	HIPPIE v2.0 HQ	HIPPIE v2.0 LQ	IWIBM HQ	IWIBM LQ
SAE [1]	0.9377	0.9224	0.8704	0.9114	0.8799
DNN-PPI [16]	0.9302	0.942	0.9414	0.9411	0.9331
Pan et al. [36]	0.9004	0.8501	NA	NA	NA
Zeng et al. [48]	0.9594	NA	NA	NA	NA
Bandyopadhyay *et al.* [49]	0.87	NA	NA	NA	NA
Yuan *et al.*	0.9762	NA	NA	NA	NA
DensePPI-CN	0.9844	0.9858	0.9886	0.9694	0.9790

Subsequently, we tested our DensePPI-PF model's performance on external standard human databases including DIP [37], HIPPIE v2.0 [38] and inWeb_inbiomap [39] compared it with standard models including DNN-PPI[16], SAE[5], GBDT[50] and Pan et al.'s own method[36] which is reported in table 4.9.

Several interspecies datasets of C.Elegans, E.Coli, H.Sapiens, M.Musculus, H.Pylori has been used to test our model's performance and measure overall generalizability. The comparison table of different standard models has been depicted in table 4.8. It has been found that DensePPI-PF model has outperformed the other standard existing classifiers.

Table 4.8: Interspecies performance comparison of our models with respect to the standard models

Dataset → Method ↓	C.Eleg (Acc %)	E.Coli (Acc %)	H.Sapi (Acc %)	M.Musc (Acc %)
DNN-XGB[34]	98.23	97.58	99.15	98.72
CNN-FSRF[1]	96.41	95.47	98.65	93.27
DPPI[51]	95.51	96.66	96.24	95.84
DeepPPI[52]	93.77	91.37	94.84	92.19
EsnDNN[53]	93.22	95.10	95.00	94.06
GcForest-PPI[54]	96.01	96.30	98.58	99.04
GTB-PPI[55]	92.42	94.06	97.38	98.08
DCT-ROF[56]	98.08	92.75	98.87	98.72
LightGBM[50]	90.16	92.16	94.83	94.57
MLD-RF[57]	87.71	89.30	94.19	91.96
MMI-NMBAC[58]	92.16	92.80	94.33	95.85
DensePPI-SC	99.85	99.73	99.90	100.00
DensePPI-PF	99.95	100.00	NA	NA

Note: NA means Not Available, as we didn't compare Human to Human PPIs and Human to Mouse PPIs.

Additionally, DensePPI-SC and DensePPI-PF model produced an accuracy of 98.83% and 99.90% on H.Pylori dataset respectively.

Table 4.9: Intraspecies performance comparison of DensePPI-CN model on standard human PPI databases

Datasets→ Methods↓	DIP	HIPPIE HQ	HIPPIE LQ	IWIBM HQ	IWIBM LQ
DNN-PPI[16]	94.33%	96.08%	93.40%	93.07%	92.80%
SAE[5]	87.73%	86.23%	81.05%	85.12%	81.87%
GBDT[59]	94.65%	94.15%	91.80%	92.84%	90.28%
Pan *et al.*[36]	88.72%	83.01%	NA	NA	NA
DensePPI -PF	100.00%	99.83%	99.52%	98.81%	98.25%

Note: NA means Not Available

One of the most important features of a deep neural network is its capacity to implicitly recognise hidden data representations. The DenseNet architecture is extremely efficient for classification of image databases. In this present work, our encoding of a protein interaction to an image and using it as an input to the proposed method produces accuracy and other metrics better than the previous classifiers as can be inferred from tables 4.4, 4.9 and 4.8. As a result, it can be concluded that the suggested DL architecture effectively extracts relevant information from the raw interaction data. When this data was utilised to classify data with DenseNet, it generated better results than existing approaches.

CHAPTER 5

Conclusion

Protein protein interaction (PPI) prediction is one of the classical problems in the domain of bioinformatics. Biological experimentally identified interactions are accurate but on the other hand is very much time consuming and costly. Therefore, there is a need to compute these PPIs *in sillico*. Also, the amount of data for negatively interacting protein pairs is inadequate and thus the previous computational approaches might have suffered from this bias.

In this thesis, deep neural networks are used to provide an unique sequence-to-image-based approach for effectively predicting protein-protein interactions (PPIs). The protein interactions and non-interactions are converted to an image and then their sub-images are generated using horizontal and vertical strides. This is done to give equal importance to every section of the PPI rather than resizing the image to match a particular size. Using non-linear transformation techniques, the deep neural network is used to extract the significant information from the raw features of the protein sequence's sub-images objectively and thoroughly. The findings of the experiments reveal that DensePPI-PF is able to accurately predict both intraspecies and interspecies PPIs. Furthermore, the suggested technique performed well on independent test sets, indicating that it can be applied to cross-species prediction. The results of the experiments show that the suggested technique is an effective tool for predicting probable protein interactions.

Although, the low prediction scores in the case of DensePPI-SC and DensePPI-PE models are due to presence of less amount of training data as reported in table 4.2. However, the cross-species prediction results of both DensePPI-PF and DensePPI-SC methods are greater than that of the *state-of-the-art* approaches.

The performance metrics for DensePPI-CN model is quite upto the mark but is not as good as the DensePPI-PF model as we have provided less data in each fold. But it is particularly helpful in systems having low system specifications. It still provides a very good result than other standard methods available till date.

However, we didn't consider any residue level biological inference while producing the colour map. Biological databases like protein data bank (PDB) may provide us with distance based measurements at the residue level which can help us greatly to guide the entire process at biological level. Also the performance evaluation has been done with the dataset with lowest complexity, measuring it on datasets with higher complexities followed by generalizing it for all the levels will lead us to achieve more accurate performance measures.

We validated our approach on the independent datasets. Our method has outperformed all standard classifiers till date on interacting protein databases. It also has produced great accuracy on DensePPI-PF model which includes interacting as well as non-interacting PPIs on benchmark data. The predicted PPIs may be useful in discovery of new drug targets, their interaction residues, therapeutic remedies and new protein functions by creating a network of proteins.

The work in this thesis can be extended by choosing the colour space in such a way so that we can incorporate biological or evolutional or functional data in the images such that we can get even higher accuracies on datasets with less amount of sub-images and we can reduce the time for training and testing the data as our approach has huge time and space requirements. A data filter may be employed to decrease the amount of training data and yet get better results.

Bibliography

[1] L. Wang, H.-F. Wang, S.-R. Liu, X. Yan, and K.-J. Song, "Predicting protein-protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[2] S. Sledzieski, R. Singh, L. Cowen, and B. Berger, "Sequence-based prediction of protein-protein interactions: a structure-aware interpretable deep learning model," *bioRxiv*, 2021.

[3] M. Chen, C. J.-T. Ju, G. Zhou, X. Chen, T. Zhang, K.-W. Chang, C. Zaniolo, and W. Wang, "Multifaceted protein–protein interaction prediction based on siamese residual rcnn," *Bioinformatics*, vol. 35, no. 14, pp. i305–i314, 2019.

[4] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017.

[5] T. Sun, B. Zhou, L. Lai, and J. Pei, "Sequence-based prediction of protein protein interaction using a deep-learning algorithm," *BMC bioinformatics*, vol. 18, no. 1, pp. 1–8, 2017.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.," *Journal of machine learning research*, vol. 11, no. 12, 2010.

[9] B. Tang, Z. Pan, K. Yin, and A. Khateeb, "Recent advances of deep learning in bioinformatics and computational biology," *Frontiers in genetics*, vol. 10, p. 214, 2019.

[10] L. Bottou *et al.*, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, p. 12, 1991.

[11] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, 1991.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] T. Moon, H. Choi, H. Lee, and I. Song, "Rnndrop: A novel dropout for rnns in asr," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 65–70, IEEE, 2015.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[16] H. Li, X.-J. Gong, H. Yu, and C. Zhou, "Deep neural network based predictions of protein interactions using primary sequences," *Molecules*, vol. 23, no. 8, p. 1923, 2018.

[17] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[18] Y.-L. Hsieh, Y.-C. Chang, N.-W. Chang, and W.-L. Hsu, "Identifying protein-protein interactions in biomedical literature using recurrent neural networks with long short-term memory," in *Proceedings of the eighth international joint conference on natural language processing (volume 2: short papers)*, pp. 240–245, 2017.

[19] S. Yadav, A. Kumar, A. Ekbal, S. Saha, and P. Bhattacharyya, "Feature assisted bidirectional lstm model for protein-protein interaction identification from biomedical texts," *arXiv preprint arXiv:1807.02162*, 2018.

[20] V. S. Rao, K. Srinivas, G. Sujini, and G. Kumar, "Protein-protein interaction detection: methods and analysis," *International journal of proteomics*, vol. 2014, 2014.

[21] Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis," in *BMC bioinformatics*, vol. 14, pp. 1–11, Springer, 2013.

[22] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4337–4341, 2007.

[23] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[24] S. Tsukiyama, M. M. Hasan, S. Fujii, and H. Kurata, "LSTM-PHV: prediction of human-virus protein–protein interactions by LSTM with word2vec," *Briefings in Bioinformatics*, 06 2021. bbab228.

[25] S. Tsukiyama, M. M. Hasan, S. Fujii, and H. Kurata, "Lstm-phv: prediction of human-virus protein–protein interactions by lstm with word2vec," *Briefings in bioinformatics*, vol. 22, no. 6, p. bbab228, 2021.

[26] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

[27] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[29] Q. Yuan, J. Chen, H. Zhao, Y. Zhou, and Y. Yang, "Structure-aware protein-protein interaction site prediction using deep graph convolutional network," *Bioinformatics*, 2021.

[30] A. K. Halder, S. S. Bandyopadhyay, P. Chatterjee, M. Nasipuri, D. Plewczynski, and S. Basu, "Juppi: A multi-level feature based method for ppi prediction and a refined strategy for performance assessment," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

[31] M. H. Schaefer, J.-F. Fontaine, A. Vinayagam, P. Porras, E. E. Wanker, and M. A. Andrade-Navarro, "Hippie: Integrating protein interaction networks with experiment based quality scores," *PloS one*, vol. 7, no. 2, p. e31826, 2012.

[32] A. K. Halder, P. Chatterjee, M. Nasipuri, D. Plewczynski, and S. Basu, "3gclust: human protein cluster analysis," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 6, pp. 1773–1784, 2018.

[33] Y. Park and E. M. Marcotte, "Flaws in evaluation schemes for pair-input computational predictions," *Nature methods*, vol. 9, no. 12, pp. 1134–1136, 2012.

[34] S. Mahapatra, V. R. R. Gupta, S. S. Sahu, and G. Panda, "Deep neural network and extreme gradient boosting based hybrid classifier for improved prediction of protein-protein interaction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[36] X.-Y. Pan, Y.-N. Zhang, and H.-B. Shen, "Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features," *Journal of proteome research*, vol. 9, no. 10, pp. 4992–5001, 2010.

[37] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, "Dip: the database of interacting proteins," *Nucleic acids research*, vol. 28, no. 1, pp. 289–291, 2000.

[38] G. Alanis-Lobato, M. A. Andrade-Navarro, and M. H. Schaefer, "Hippie v2. 0: enhancing meaningfulness and reliability of protein–protein interaction networks," *Nucleic acids research*, p. gkw985, 2016.

[39] T. Li, R. Wernersson, R. B. Hansen, H. Horn, J. Mercer, G. Slodkowicz, C. T. Workman, O. Rigina, K. Rapacki, H. H. Stærfeldt, *et al.*, "A scored human protein–protein interaction network to catalyze genomic interpretation," *Nature methods*, vol. 14, no. 1, pp. 61–64, 2017.

[40] P. Green-Armytage, "A colour alphabet and the limits of colour coding," *JAIC-Journal of the International Colour Association*, vol. 5, 2010.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[43] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *arXiv preprint arXiv:1507.06228*, 2015.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[46] F. Chollet *et al.*, "Keras," 2015.

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[48] J. Zeng, D. Li, Y. Wu, Q. Zou, and X. Liu, "An empirical study of features fusion techniques for protein-protein interaction prediction," *Current Bioinformatics*, vol. 11, no. 1, pp. 4–12, 2016.

[49] S. Bandyopadhyay and K. Mallick, "A new feature vector based on gene ontology terms for protein-protein interaction prediction," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 14, no. 4, pp. 762–770, 2016.

[50] C. Chen, Q. Zhang, Q. Ma, and B. Yu, "Lightgbm-ppi: Predicting protein-protein interactions through lightgbm with multi-information fusion," *Chemometrics and Intelligent Laboratory Systems*, vol. 191, pp. 54–64, 2019.

[51] S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, "Predicting protein–protein interactions through sequence-based deep learning," *Bioinformatics*, vol. 34, no. 17, pp. i802–i810, 2018.

[52] X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, and Y. Zhang, "Deepppi: boosting prediction of protein–protein interactions with deep neural networks," *Journal of chemical information and modeling*, vol. 57, no. 6, pp. 1499–1510, 2017.

[53] L. Zhang, G. Yu, D. Xia, and J. Wang, "Protein–protein interactions prediction based on ensemble deep neural networks," *Neurocomputing*, vol. 324, pp. 10–19, 2019.

[54] B. Yu, C. Chen, X. Wang, Z. Yu, A. Ma, and B. Liu, "Prediction of protein–protein interactions based on elastic net and deep forest," *Expert Systems with Applications*, vol. 176, p. 114876, 2021.

[55] B. Yu, C. Chen, H. Zhou, B. Liu, and Q. Ma, "Gtb-ppi: Predict protein–protein interactions based on l1-regularized logistic regression and gradient tree boosting," *Genomics, proteomics & bioinformatics*, vol. 18, no. 5, pp. 582–592, 2020.

[56] L. Wang, Z.-H. You, S.-X. Xia, F. Liu, X. Chen, X. Yan, and Y. Zhou, "Advancing the prediction accuracy of protein-protein interactions by utilizing evolutionary information from position-specific scoring matrix and ensemble classifier," *Journal Of Theoretical Biology*, vol. 418, pp. 105–110, 2017.

[57] Z.-H. You, K. C. Chan, and P. Hu, "Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest," *PloS one*, vol. 10, no. 5, p. e0125811, 2015.

[58] Y. Ding, J. Tang, and F. Guo, "Predicting protein-protein interactions via multivariate mutual information of protein sequences," *BMC bioinformatics*, vol. 17, no. 1, pp. 1–13, 2016.

[59] Y.-H. Qu, H. Yu, X.-J. Gong, J.-H. Xu, and H.-S. Lee, "On the prediction of dna-binding proteins only from primary sequences: a deep learning approach," *PloS one*, vol. 12, no. 12, p. e0188129, 2017.