



A Survey on Deep Networks Approaches in Prediction of Sequence-Based Protein–Protein Interactions

Bhawna Mewara¹ · Soniya Lalwani¹

Received: 9 November 2021 / Accepted: 6 May 2022 / Published online: 19 May 2022
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

Abstract

The prominence of protein–protein interactions (PPIs) in system biology with diverse biological procedures has become the topic to discuss because it acts as a fundamental part in predicting the protein function of the target protein and drug ability of molecules. Numerous researches have been published to predict PPIs computationally because they provide an alternative solution to laboratory trials and a cost-effective way of predicting the most likely set of interactions at the entire proteome scale. In recent computational methods, deep learning has become a buzzword with numerous scientific researches. This paper presents, for the first time, a comprehensive survey of sequence-based PPI prediction by three popular deep learning architectures i.e. deep neural networks, convolutional neural networks and recurrent neural networks and its variants. The thorough survey discussed herein carefully mined every possible information, can help the researchers to further explore the success in this area.

Keywords Deep learning · Deep networks · Protein–protein interactions · Recurrent neural network · Long short term memory

Abbreviations

AA	Amino acid	DNA	Deoxyribonucleic acid
AC	Auto-covariance	DNN	Deep neural network
ACC	Autocross-covariance	DT	Decision tree
AdaBoost	Adaptive boosting	DWT	Discrete wavelet transform
Adam	Adaptive moment estimation	E-ELM	Ensemble extreme learning machine
AE	Auto-encoder	GPU	Graphics processing unit
AF	Activation function	GRU	Gated recurrent units
AUC	Area under curve	KNN	K-nearest neighbor
BRNN/ BiRNN	Bidirectional recurrent neural networks	LD	Local descriptors
CNN	Convolutional neural network	LR	Learning rate
CT	Conjoint triad	LSTM	Long short term memory
CV	Cross-validation	MCD	Multi-scale continuous and discontinuous
CWT	Continuous wavelet transform	MLD	Multi-scale local descriptor
DBN	Deep belief network	MLP	Multilayer perceptron
DIP	Database of interacting proteins	NB	Naïve Bayes
DL	Deep learning	NLP	Natural language processing
DN	Deep networks	NN	Neural network
		PCA	Principal component analysis
		PPI	Protein–protein interaction
		PR	Precision-recall
		RBM	Restricted Boltzmann machine
		RCNN	Residual recurrent convolutional neural network
		ReLU	Rectified linear unit

✉ Bhawna Mewara
mewara.bhawna2203@gmail.com

Soniya Lalwani
slalwani.math@gmail.com

¹ Department of Computer Science and Engineering, Career Point University, Kota, India

RF	Random forest
RNA	Ribonucleic acid
RNN	Recurrent neural network
ROC	Receiver operating characteristic
SAE	Stacked auto-encoder
SGD	Stochastic gradient descent
SVM	Support vector machine

Introduction

Proteins are essential to organisms and participate in every process virtually within cells. Despite the wide range of functions, all proteins are made out of the same twenty-one building blocks called amino acid (AAs), but combined in different ways. AAs are made of carbon, oxygen, nitrogen, and hydrogen and some contain sulphur atoms. These atoms form amino groups, a carboxyl group, and a side chain attached to a central carbon atom as shown in Fig. 1. The side chain determines the AA's properties and this is the only part that varies from one AA to another AA.

Two AA molecules can be covalently joined to a substituted amide linkage termed as peptide bond and it returns a Dipeptide [1]. Such a linkage is formed by the removal of the elements of water *i.e.* dehydration from the alpha-carboxyl group of one AA and alpha-amino group of another AA as depicted by Fig. 2. Similarly, three AAs can be joined by two peptide bonds to form tripeptide and four to form tetrapeptide, and so on. When many AAs are joined in this fashion, the product is called a polypeptide. An AA in a peptide is often called a residue *i.e.* the part left over after losing the water. Protein may have 1000 s of AA residues. Generally, the terms protein and polypeptide are used interchangeably. Molecules referred to as Polypeptide have a molecular weight (MW) below 10,000 daltons and those called proteins have higher MW.

Proteins usually do not function alone, they need a partner to accomplish their functions. The partner may be DNA,

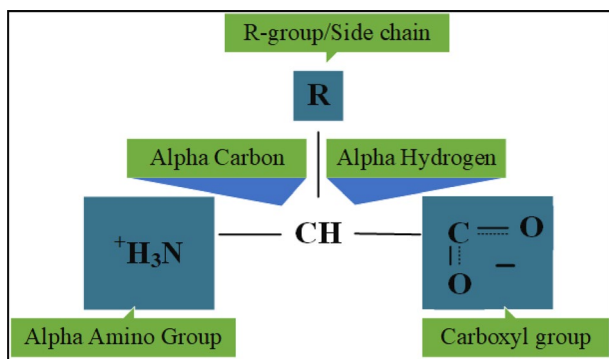


Fig. 1 Structure of amino acid

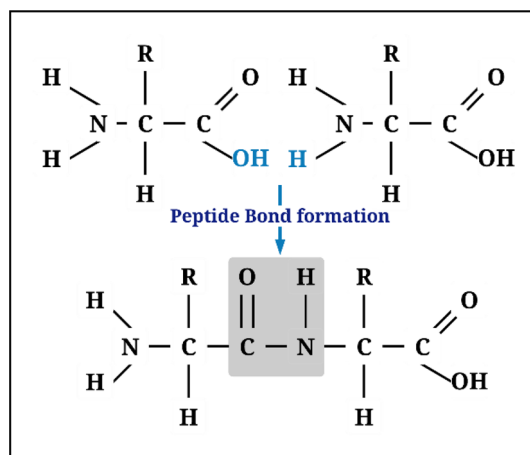


Fig. 2 Formation of peptide bond

RNA, or proteins. If a single protein is present inside the cell it is not that functional but together all the proteins are functioning with themselves. And if a protein interacts with another protein, or if two or more proteins are cross-talking with each other by some signaling processes, it is termed as protein–protein interactions (PPI) [2]. Protein control and mediate many of the biological activities of the cell by these interactions. For e.g. Muscle contraction (is possible due to PPI between active myosine filaments), cell signaling, cellular transport (molecule coming out and going inside the cell using PPI) [3]. So PPIs play a vital role in many cellular processes.

However, disruption or formation of abnormal interactions can lead to a disease state. This drives many researchers to predict PPI at the early stages of the disease symptoms. As some of the diseases show their symptoms in the later stage of the disease which may be lead to complexity in medication or may be deadly. Prior information about PPIs can offer a clear vision to detect drug targets, further biological processes, and new remedies for diseases [3]. Compared to the investigational methods, such as tandem affinity purifications (TAP) [4], protein chips [5], and efficient biological methods, computational approaches are revealing better exposure for PPIs prediction, as they are less time-consuming and more proficient [6].

Machine learning (ML) methodologies to predict PPIs govern most of the computational methods [7, 8]. Framing a suitable feature set and selecting favorable machine learning algorithms are two major stages for prosperous predictions. The feature set can be constructed wisely in such a way that they could cover the maximum information or key features from the structure of the proteins. Among the structures, the primary structures *i.e.* the sequences of the protein are the most common to work on because of the huge data availability [9]. Several feature extraction methods have been developed in the past for representing the

protein information in numerical form that are widely used to possibly extract protein interaction information [10–15]. For the PPIs prediction purpose, each feature extraction algorithm requires a favorable classifier to appropriately classify the interaction or no interaction according to the feature sets. Various classification algorithms have been developed like RF, SVM and their derivatives [16], gradient boosting decision trees [17], and ensemble classifiers [18].

Recently, DL technology has come into the limelight with numerous scientific researches that help in many applications like image recognition [19], speech recognition [20], machine language translation [21], computer vision [22], and many more. In DL, specifically, DNNs, RNNs and CNNs have contributed a lot in real-life applications and ease human efforts. Numerous noteworthy DL-based researches are being published in the field of bioinformatics [23, 24].

This paper focuses on some DL approaches using in the PPI prediction task, in the successive sections, a short name is used as deep networks (DNs) to represent DNNs, CNNs and RNNs and its variants.

The aim of this paper is to provide a comprehensive survey of DN applications in the field of PPI prediction. In this review, the recent progress in applying DN techniques to the problem of PPI prediction is summarized and discussed the possible pros and cons. The scope of this paper is limited to the primary structure of the protein *i.e.* the sequence-based PPI prediction with DNs. The significance and the approaches to represent protein sequence based on DN are discussed for the first time. The central importance of proteins' primary structure is also emphasized.

Therefore, the paper is organized as follows: “[Introduction](#)” section presents the outline about the protein, importance of PPI, several methods to detect PPI, and recent advancement of computational approaches in the field of Bioinformatics. “[Outline of Deep Networks](#)” section familiarizes the concept of DNs and how DNs can be proved beneficial in PPI prediction. “[Approaches for sequence-based Protein–Protein Interaction Prediction using Deep Networks](#)” section illustrates the various research publication of sequence-based PPI prediction using DNs along with their pros and cons and performance achieved. “[Implementation of Cited Papers](#)” section presents the manual implementation of cited papers. In the succession to analyze the adeptness of DNs in PPI prediction, a fair comparison is made in “[Comparison with State-of-the-art Methods](#)” section with State-of-the-art methods. At last, the paper is concluded with future aspects in this area. This review is focused to help both computational biologists to achieve familiarity with the DN methods applied in protein modeling, and computer scientists to expand perspective on the biologically significant problems that may help from DL methods.

Outline of Deep Networks

Deep learning architecture can be understood as the ANNs with several layers and researchers have contributed several types of DL architectures based on the considered input and purpose of the particular research. This review mainly considers three DL architectures: DNNs, CNNs and RNNs. However, several researchers included all DL architectures in DNNs [25, 26]. This paper considers ‘DNNs’ to discuss specifically SAE [27] which use AEs [28] as the elementary units of NNs [29]. The reason behind these considerations is the limited scope of this paper which mainly focuses to deliver the significance of DNs using sequential information of the input data of PPI for the prediction task.

Generally in DL architectures, there are two principle elements that lift up the performance: Optimization and Regularization. The target during training is to optimize the weight parameters in each layer so that the important and relevant features can be learned from the input by filtering out the irrelevant information and transfer an abstract form or reduced number of features to the next layer. The optimization procedure follows an algorithm to update the weight parameters based on the SGD [30]. Regularization is a process to evade over-fitting problem which usually occurs while training. Some regularization processes have been developed like weight decay [31], Dropout [32], rnnDrop [33]. Recently, a novel regularization technique has been proposed [34], which operates in batches by doing the normalization of features.

The following part of this section gives a brief knowledge about three DL approaches DNNs, RNNs and CNNs that have greatly contributed to the prediction task of PPIs using sequential information only.

Deep Neural Networks

A DNN, in simple words, is a network that is deep *i.e.* which has many hidden layers along with the input layer and an output layer as shown in Fig. 3. For the given input data, the outputs are sequentially calculated with the layers of the network. The input vector at each layer includes the output of the previous layers' unit which are then multiplied by the weight vector of the considered layer that resulted in the weighted sum. The output of a particular layer is computed by applying some non-linear function (ReLU, sigmoid, etc.) [35] to the weighted sum which results in more abstract representations from the previous layer output as follows [36]:

$$p_x^{(O+1)} = \mu(w_x^{(O+1)}p^O + z_x^{(O+1)}) \quad (1)$$

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

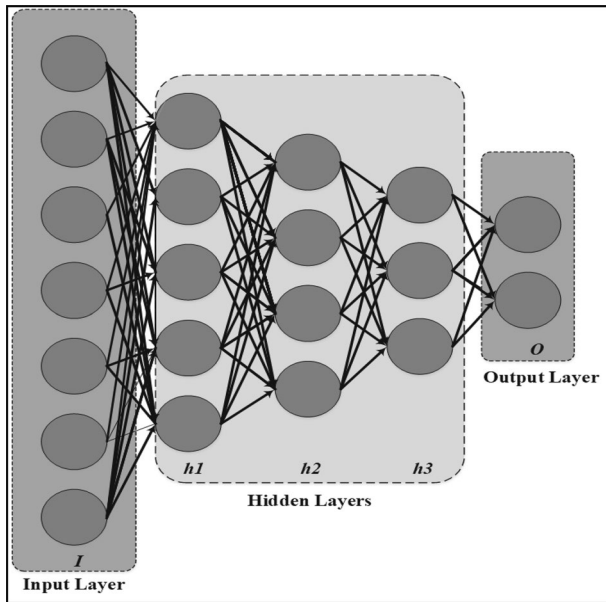


Fig. 3 Basic structure of DNNs with input units I , three hidden units $h1$, $h2$ and $h3$, in each layer and output units O . At each layer, the weighted sum and non-linear function of its inputs are computed to obtain an abstract representation

DNNs work very well for scrutinizing high-dimensional data. Good researches in bioinformatics cannot be completed with small data, therefore the data available in this field is usually high-dimensional and complex and thus DNNs guarantee favorable opportunities for the researchers to work in. DNNs have the potential to give knowledge to more readily comprehend by extract the highly abstract and related information from the data. Though the raw data is the only requirement for DNNs to learn graded features, manually crafted features have frequently been given as contributions. This concludes that the abilities of DNNs have not yet completely been taken advantage of. It is believed that the future advancement of DNNs in bioinformatics will come from examinations concerning appropriate approaches to encode crude information and take in reasonable features from them.

Recurrent Neural Networks

The structure of RNNs has a recurring link in each hidden layer which is responsible to operate sequential information by some recurrent computation as shown in Fig. 4. The previous output (state vector) is kept in the hidden units and for the current state, the output is calculated using the previous state vector and the considered input [37]. The following two equations express the evolvement of RNN over time [38]:

$$O_t = \delta(h_t; \theta) \tag{2}$$

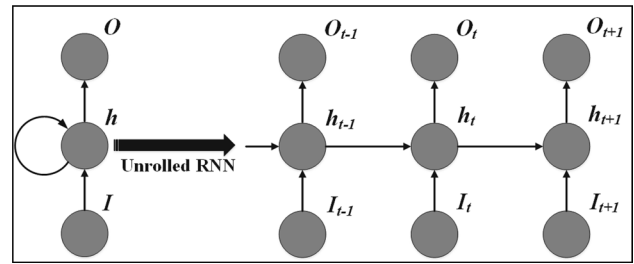


Fig. 4 Basic structure of RNNs with an input unit I , a hidden unit h and an output unit O . The recurrent computation can be expressed more explicitly if the RNNs are unrolled in time. The index of each symbol represents the time step. In this way, h_t receives input from I_t and h_{t-1} and then propagates the computed results to O_t and h_{t+1}

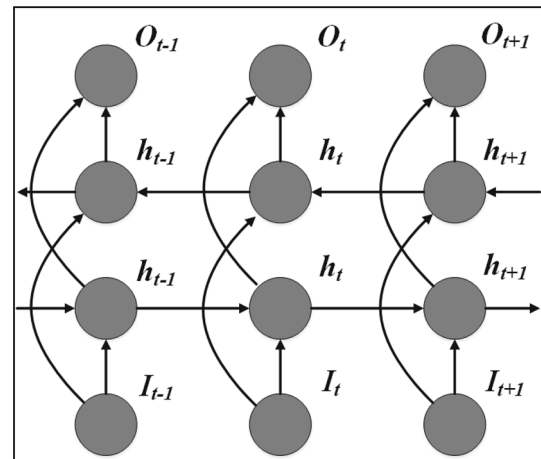


Fig. 5 Basic structure of BRNNs unrolled in time. For each time step, there are two hidden layers. The information from both hidden units is propagated to O_t

$$h_t = g(h_{t-1}, I_t; \theta) \tag{3}$$

here, θ includes weights and biases for the network, the first equation express the dependency of the output O_t at time t only with the hidden layer h_t using some computation function δ and the second equation shows the dependency of the hidden layer h_t at time t with that of h_{t-1} at time $t-1$ and the input I_t at time t .

RNNs specifically BRNNs are popularly used in applications where previous information is required for the current output (as shown in Fig. 5) like speech recognition, Google translator, etc. The appearance of RNN structure is simpler than DNNs in terms of the number of layers, but if the structure of RNN is unrolled with time, it is even deeper.

Though, this leads to two popular hindrances: vanishing gradient and long-term dependencies, researchers have been overcome these issues by adding some complex units and develop some variants of RNNs, like LSTM, GRU. Today, RNNs have been utilized effectively in numerous domains

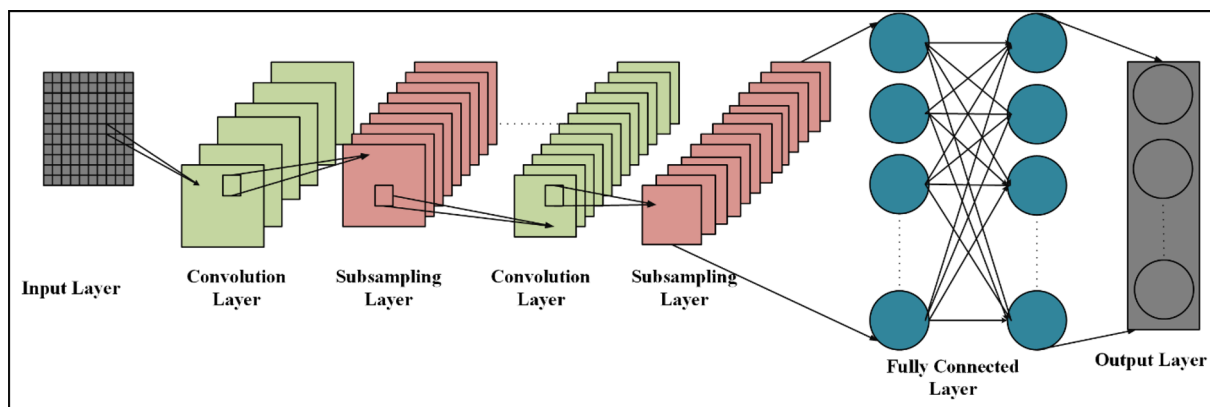


Fig. 6 The baseline structure of CNN

including NLP and language interpretation [39–42]. The nature of identifying the PPI is practically identical to the modeling tasks undertaken in researches of NLP as the two of them intended to analyze the shared impact of two arrangements dependent on their underlying features. Proteins are reported in groupings with a more preserving manner, also a bigger scope of lengths. Therefore, accurately covering the PPI not only requires significantly more extensive learning to strain the important and relatable features from the whole sequences but also retain the long-term ordering information. If the PPI prediction task and the working of considered DNs are carefully observed, then it can be concluded that these DL architectures can contribute a lot to the considered prediction tasks and could be the emerging area for researchers.

Convolutional Neural Network

Convolutional neural network is a branch of Deep Learning algorithm which can take an input in the form of image, allocate learnable weights and biases to various features of the image and be able to distinguish one from the other with the minimum pre-processing requirement as compared to other classification algorithms [43]. The structure of CNN is basically a feed-forward neural network whose neurons can retort to the nearby units in a part of the coverage and have outstanding performance for data feature extraction [44]. The output value is computed using forward propagation and weights and biases are adjusted using back propagation. Figure 6 shows the structure of CNN comprises of the input layer, the convolutional layer, subsampling layer, full connection layer and the output layer.

The feature map M_l at l th layer is computed as [44]:

$$M_l = f(M_{l-1} \circ w_l + b_l), \tag{4}$$

where w_l is the weight matrix of the convolution kernel of l th layer, b_l means the offset vector, f represents the activation

function and operator \circ denotes convolution operations. The subsampling layer usually behind the convolutional layer and the feature map is sampled according to given rules. Suppose, M_l is a subsampling layer, its sampling formula is:

$$M_l = \text{subsampling}(M_{l-1}). \tag{5}$$

The fully connected layer is responsible for classification of the extracted features via several convolution and sub sampling operations. The fundamental mathematical notion of CNN is to map the input matrix M_0 to a new feature representation R through multi-layer data transformation.

$$R(l) = \text{Map}(C = c_l | M_0; (w, b)) \tag{6}$$

where c_l represents the l th label class, M_0 denotes the input matrix, and R denotes the feature expression. The goal of CNN training is to minimize the network loss function $R(w, b)$. At the same time, to ease the over-fitting problem, the final loss function $Z(w, b)$ is usually controlled by a norm, and the intensity of the over-fitting is controlled by the parameter ϵ .

$$Z(w, b) = R(w, b) + \frac{\epsilon}{2} w^T w. \tag{7}$$

Numerous research papers have been published in the discussed domain. In the next section, the related papers are briefly discussed along with their objectives, approaches, considered dataset, and performance measures.

Approaches for Sequence-Based Protein–Protein Interaction Prediction Using Deep Networks

To the best of our knowledge, to date, there are around 30 research papers have been published for PPI prediction using DNs that are using sequence information as input.

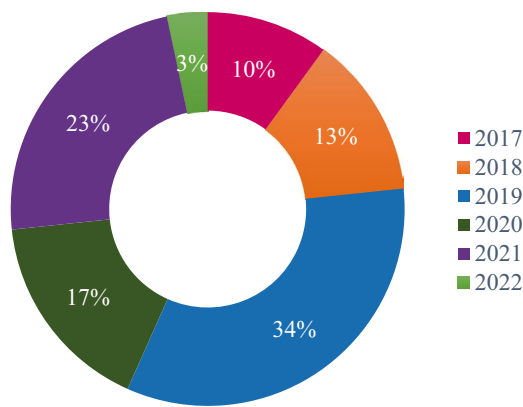


Fig. 7 Publication analysis of PPI prediction approaches using DNs

The same is also depicted by the publication analysis of sequence-based PPI prediction using DNs in Fig. 7. This section details all the studies performed on PPI prediction tasks using DNs so far. The summary of the same is also provided in Table 2. Out of 30, four papers are based on identifying PPIs using biomedical text dataset which is a part of the Biomedical Natural Language Processing (BioNLP) [45] community, and the remaining are using physical protein pair interaction datasets. Therefore, the studies are classified on the basis of: year of publication; Research objectives; Approach to predict PPIs; Types of the dataset used; and Hyperparameters of the network. The term ‘Strategy’ written after each section is used to indicate the category of approach in the table. All the important abbreviated terms of the table are provided in expanded form in the corresponding text, whereas the basic abbreviations are provided after the abstract. The detailed description of this section is broadly divided on the basis of the dataset used. For better understanding, an abbreviated form mentioned in Table 1 is used for the dataset considered by the cited paper in subsequent sections.

Table 1 Short names given for datasets considered by cited papers

S. No	Dataset	Short Name	S. No	Dataset	Short Name
1	AiMed	<i>A</i>	11	<i>H. pylori</i>	<i>k</i>
2	<i>Arabidopsis thaliana</i>	<i>B</i>	12	<i>H. sapiens</i>	<i>l</i>
3	<i>B. subtilis</i>	<i>C</i>	13	HPRD50	<i>m</i>
4	<i>B. taurus</i>	<i>D</i>	14	IEPA	<i>n</i>
5	<i>Bacillus anthracis</i>	<i>e</i>	15	LLL	<i>o</i>
6	BioInfer	<i>f</i>	16	<i>M. musculus</i>	<i>p</i>
7	Benchmark Dataset	<i>g</i>	17	<i>R. norvegicus</i>	<i>q</i>
8	<i>C. elagan</i>	<i>h</i>	18	<i>S. cerevisiae</i>	<i>r</i>
9	<i>Drosophila melanogaster</i>	<i>i</i>	19	<i>S. pombe</i>	<i>s</i>
10	<i>E. coli</i>	<i>j</i>	20	<i>Yersinia pestis</i>	<i>t</i>

Benchmark Dataset: 2010 HPRD, the 2010 HPRD NR, the DIP (Human), HIPPIE, inWeb_inbiomap

Prediction Using Paired Protein Interaction Dataset

Some scholars proved that the DNs are capable enough to capture the potential features from the input protein raw data while some researchers include the hand-crafted features with DNs to enhance the performance of PPIs prediction tasks. Therefore, this sub-section is again categorized according to the inclusion and exclusion of manual feature engineering.

Strategy-A: Inclusion of Manually Crafted Features

The most important factor to develop a computational technique for the prediction of PPIs is to mine extremely preferential features that can well define proteins. Several publications proposed novel methods for representing the protein information in numerical ways as shown in the Table 3 which are popularly used by several publishers to produce proficient methods that can extract the protein interaction information more finely.

The use of DL algorithms in sequence-based PPIs prediction task began from 2017 [46] by proposing the use of SAE to filter the heterogeneous features in the low-dimensional space. The protein sequences were numerically represented using AC and CT methods which were then fed to the model for training with tenfold CV. The author observed that with a one-hidden layer, both the AC model having 400 neurons and the CT model with 700 neurons attained the best performances and concluded that the prediction performances of the model do not depend on the number of neurons and layers. Then for the final model construction, they took AC because of its better performance and trained with the entire benchmark dataset, finally compared the results with the previous ML approaches that used the same dataset. Following the similar pattern, Du et al. [47] employed five widely used descriptors to represent protein sequence which is then effectively learned by a DNN model named DeepPPI. The author later showed the performance of DeepPPI using two

Table 2 Publication analysis of DN approaches in prediction of sequence-based PPIs

Year [References]	Research objective	Approach	Considered dataset/corpora	Hyper parameters	Highest reported accuracy (in %)
<i>STRATEGY-A</i>					
2017 [46]	Showed the effectiveness of PPI prediction by applying DL algorithm for the very first time (as per the article)	AC and CT+SAE Chosen AC for final model design Evaluation: tenfold CV	g ; SSD: j , i and h	The AC model with 400 neurons and the CT model with 700 neurons, having one-hidden layer in both	97.19 (g)
2017 [47]	Aim to improve PPI prediction performance by effectively learning the representations of proteins from Common protein descriptors	5 Descriptors-AAC; Dipeptide Composition; Composition; Transition and Distribution; Quasi-Sequence-Order Descriptors; Amphiphilic Pseudoamino Acid Composition; Used two separate DNN model (Difference in the input structure) Evaluation: fivefold CV	r , k , and l ; SSD: l , p , k , h and j	LR: 0.01 Batch size: 64 Momentum rate: 0.9 Adaptive LR parameter- SGD AF- ReLU Dropout rate - 0.2	98.14 (l)
2019 [48]	Considered the order relationship of the entire amino acid sequence and time constraints issues	Proposed a sequence-based method based on a novel representation of the matrix of sequence (MOS), then used DNN and combine to predict PPI	G	the number of hidden layer nodes—64, AF: ReLU, Adam optimizer, batch size: 128, dropout: 0, LR: 0.01	94.34
2019 [49]	Accurately predicting PPI based on the properties of AA on the protein primary sequences	Conjoint AAindex modules descriptor (CAM), ensemble of hybrid deep neural networks (fully connected recurrent neural network) Evaluation: tenfold CV	l and r	Input features: 343*6 LR: 1.4792765037709115E-5 L2 regularization: 1.978894473010271E-5 ADAM 6 layers, Dropout rate: 0.9	94.72 (r)
2019 [50]	Discover effective feature, underlying patterns and inherent mappings	Used two separate DNN modules to squeeze out latent features from two embedding vectors obtained by Res2vec method of representation learning Evaluation: fivefold CV	l and r ; SSD: l , p , k , h and j	Residue dim: 20 Window size: 4 Protein length: 850 N/w depth: 4	98.71 (l)
2019 [52]	Predict PPIs based on different representations of amino acid sequences	AC, LD, MCD; 9 independent DNNs having different parameter settings final two-layered NN for the prediction Evaluation: fivefold CV	r ; SSD: l , p , k , h and j	Different parameter acc. To the considered features. With ADAM optimizer and ReLU AF	95.29 (r)

Table 2 (continued)

Year	[References]	Research objective	Approach	Considered dataset/corpora	Hyper parameters	Highest reported accuracy (in %)
2019	[53]	PPI prediction to resolve operational time issue, and large costs as well as low prediction accuracy	ProtVec and protein signatures methods LSTM architecture	VCP protein data of ' <i>k</i> ' taken from BioGRID	Input seq length = 400 4 1D convolutional layers, 4 pooling layers with average pooling, 1 LSTM layer and 1 fully connected layer with Softmax with 1024 neurons N/A	92 (<i>k</i> using ProtVec)
2019	[56]	Extract deeply hidden protein feature and remove redundant information to enhance the prediction results	Used CNN for feature extraction and Feature selective rotation forest (FSRF) for noise elimination	<i>r</i> and <i>k</i> ; SSD: <i>e</i> , <i>h</i> , <i>j</i> and <i>n</i>	N/A	97.75 (<i>r</i>)
2020	[57]	Optimize the predictive performance of PPI	AC,CT,LD Build DNN model and use dropout method	<i>g</i> ; SSD: <i>h</i> , <i>i</i> , <i>j</i> , <i>l</i>	LR: 0.001 Batch size: 128 AF: ReLU Adam optimizer Cost function: Cross-entropy Dropout: multiple Batch size: 128, LR: 0.05	98.60 (<i>g</i> : CT with dropout 0.3)
2020	[58]	Improve the variety of the features to be considered in the prediction, insure the training time caused on the extra training either a backward one or a forward one	Physicochemical properties and then applied DWT and CWT with 25-scale mexh wavelet function; "Y-type" NN model, comprising a weight-sharing Bi-RNN layer, a buffer layer, and a dense layer Evaluation: fivefold CV	<i>g</i> ; SSD: <i>l</i> , <i>p</i> , <i>k</i> , <i>h</i> and <i>j</i>	Cost function: Cross-entropy Dropout: multiple Batch size: 128, LR: 0.05	99.57 (<i>g</i>)
2020	[59]	Employed multimodal information that integrates sequence-based and 3D structural information of proteins to improve prediction capability	AC and CT for sequence-based, Used ResNet50 for structural-based; LSTM classifier Evaluation: threefold CV	<i>l</i> and <i>r</i>	N/A	97.20 (<i>l</i>)
2020	[60]	Considered the issues of gradient-descent learning to the optimum global solution with increasing network size	CT; SAE and ML-ELM DNN for prediction Evaluation: fivefold CV	Indonesian Herbal Medicine- Herbs Analytics, STRING-DB	N/A	89 (Herbs with SAE)
2021	[61]	Hybrid method is an effective tool to accurately predict potential protein interactions	AAC, CT, LD; The DNN extracts the hidden information through a layer-wise abstraction from the raw features that are passed through the XGB classifier Evaluation: fivefold CV	Four standard intraspecies (<i>r</i> core and full, <i>k</i> , <i>l</i>) and two standard interspecies datasets (Human host with <i>e</i> and <i>t</i> pathogen)	For DNN LR: 0.01 Batch size: 64 Momentum Rate: 0.9 AF: ReLU Dropout rate: 0.2 Loss function: binary_crossentropy	98.35 (<i>r</i>)

Table 2 (continued)

Year [References]	Research objective	Approach	Considered dataset/corpora	Hyper parameters	Highest reported accuracy (in %)
2021 [72]	Identify protein functions faster	Used AVL tree for numerical representation; 3-layered BiRNN with ReLU AF; followed by Flatten, Batch normalization and Dropout function; next two FC layer Evaluation: tenfold CV	From NCBI and BioGrid database	Dropout- 0.25 1FC = 512 neurons 2 FC—for prediction with sigmoid func SGD optimizer, LR-0.0001 Momentum = 0.9 binary crossentropy for model loss with 500 epochs	Multiple
2021 [64]	The impact of integrating multiple features that are used in the prediction of PPIs either alone or integrated with some other features	43 features generated by three different methods: 22 Evolutionary features based on generation of a PSSM using PSI-Blast algorithm, 17 structural features generated via a DL model SPIDER2, 7 features generated by popularly used physiochemical properties SAE used as classifier Evaluation: tenfold CV	L	Input-92 Unit- 75 AF-sigmoid LR-1 Momentum = 0.5	83.55 (l)
2021 [68]	Address the problem of protein-protein interaction by employing AEs solely	AC, CT; An ensemble of two AEs three types of NN architectures were used: Joint-Joint architecture, Siamese-Joint architecture; Siamese-Siamese architecture Evaluation: tenfold CV	g ; SSD: h, i and j	Two layered encoder having 600 neurons linked to a bottleneck layer of 300 neurons, followed by a symmetric decoder Selu AF, ADAM optimizer, Initial LR- 0.0005, batch size- 64 for 2000 epochs N/A	97.9 (g)
2021 [69]	Improving prediction performance	A new feature extraction method called MSR based on the spectral radius and BLOSUM62 matrix; AC was applied to supplement and extract effective sequence information; The GRNN used as the classifier Evaluation: tenfold CV	r, l and k ; SSD: m, i, h, p, k and l Two significant PPI networks (CD9 and Wnt)		99.97 (k)
2021 [73]	Proposed a data encoding method Sequence-Statistic-Content (SSC) for feature extraction to enhance precision by providing more features with extra information	Protein sequence encoded by three-channel format using statistical information. Then 2D CNN is used with SSC encoded features for prediction task	r, j, i and b	4 conv. Layer Dropout: 0.25 AF: Leaky ReLU	78.40 (b)

Table 2 (continued)

Year [References]	Research objective	Approach	Considered dataset/corpora	Hyper parameters	Highest reported accuracy (in %)
STRATEGY-B					
2018 [74]	Investigated the capability of auto feature engineering	Embedding layer + 3-layered CNNs, LSTM + FC Evaluation: fivefold CV	g ; SSD: j, i, h and p	Input length = 1200 Batch size = 128 3 Conv layer with filter length = 10 and ReLU, 3 Max-Pooling layer LSTM Layer Adam optimizer	98.78 (g)
2018 [75]	The features are learned through an optimization process, leveraging the increasing amount of available PPI data	Tokenization, embedding layer, a recurrent layer (with GRU units) and a fully connected layer each for two branches, Branch normalization, Dropout layer Evaluation: fivefold CV	r, k and l	Employed the categorical cross entropy, paired with the RMSProp gradient descent optimization algorithm Input length-1000 Embedding- 512 features RNN Output dimension = 64	97.98 (l)
2018 [76]	Leveraging existing high-quality experimental PPI data and evolutionary information of a protein pair under prediction	Employed three modules: Convolutional module (convolutional layer, ReLU, batch normalization, and pooling layer) Random Projection module (2 FC sub-networks) Prediction Module (perform element-wise multiplication to calculate probability score) Evaluation: tenfold CV	l and r SSD: $l, r, p, i, h, j, b, c, d, s$ and q	Multiple	94.55 (r)
2019 [77]	Focus on both robust local features and contextualized information, which are significant for capturing the mutual influence of proteins sequences Address three ppi prediction tasks: Interaction prediction, estimation of binding affinity and prediction of interaction type	Deep Siamese architecture of residual RCNN: convolution layers with pooling and bidirectional residual gated recurrent units Evaluation: fivefold CV	r and l	N/A	97.09 (r)
2019 [79]	Compare two carefully designed deep learning models and show pitfalls to avoid while predicting PPIs	Compared two DL models: a FC model and a recurrent model intended to show the downsides which are needed to avoid while predicting PPIs	L	Multiple parameters Common in both models: loss function: binary cross-entropy, Adam optimizer, LR: 0.001	Multiple

Table 2 (continued)

Year [References]	Research objective	Approach	Considered dataset/corpora	Hyper parameters	Highest reported accuracy (in %)
2020 [80]	Efficient computation performance to accelerate PPI prediction	Embedding method to represent AA sequences; Powerful feature extraction using proposed ResNet algorithm; ResPPI algorithm is a combinatorial process of five residual units and each residual unit comprises of: three 2D convolution layers each followed by batch normalization and then a mapping function and ReLU; FC layer having a softmax function for binary classification Evaluation: fivefold CV	g	LR-0.001 Batch size-32 Input length -128 2 Conv. Layer- 32 2 Conv. Layer-64 ReLU Softmax for prediction Binary cross-entropy for loss minimization	96.69
2021 [83]	Generalizes better to new species and is robust to limitations in training data size Checks for compatible residual responsible for interaction in two proteins	Protein Embedding Projection module; Contact module; Interaction module Evaluation: fivefold CV	i ; SSD: i, p, r, h and j	Projection Dimension = 100, a hidden Dimension = 50, a convolutional filter with width $2w+1=7$, and a local max-pooling width = 9 Weights were initialized using PyTorch defaults. Batch size = 25, the Adam optimizer with a LR of 0.001, and trained all models for 10 epochs LR: 0.001 AMSGrad optimizer	Multiple
2022 [85]	Used mask multi-scale CNN to contribute in prediction enhancement by providing additional insights into each input neuron	Numerous convolution filters arranged in parallel fashion to extract deeper and refined protein features from the profiles. Also employed single-protein class and masking operation	l and r ; SSD: h, i and j		98.12 (l)
STRATEGY-C					
2017 [88]	Identify PPIs in biological literature / Incorporate linguistic and semantic information	Embedding layer, Bi-RNN, FC Evaluation: tenfold CV and cross-corpus (CC)	a and f	Embedding- 200, LSTM-400, RMSProp optimizer Dropout rate- 0.5	P 87 R 87.4 F-1 87.2 (f)

Table 2 (continued)

Year [References]	Research objective	Approach	Considered dataset/corpora	Hyper parameters	Highest reported accuracy (in %)
2018 [89]	Efficient information extraction from the large collection of biomedical texts for PPI identification	A Shortest Dependency Path (SDP) was created to interpret more relevant information using a Bi-directional LSTM (Bi-LSTM) Part-of-Speech (POS) and Position features were also explored Evaluation: tenfold CV	a and f	Number of LSTM units 64 Dropout rate-0.3 Sigmoid—ADAM Optimization algorithm Adam Epochs-130 Size of MLP layer output-30	P 91.1 R 82.2 F-1 86.45 (a)
2019 [92]	Introducing attention mechanism to pay more attention to the most influential segments of texts for a relationship category	Underlying architecture is same as 6 with minor changes: include an attention layer and used a stacking strategy in the Bi-LSTM unit	a, f, m, n and o	Number of LSTM units 64 Dropout ratio 0.3 Activation function Sigmoid Optimization algorithm Adam Epochs (AiMed & BioInfer) 115 Epochs (HPRD50, IEPA & LLL) 50 Size of MLP layer output 30 No. of LSTM layers 6 Context vector size 75	P 93.96 R 92.63 F-1 93.29 (a)
2019 [94]	Identify PPI from bio-medical text	Traversed the PPI-related sentences through the network topology of tree-like structure in such a way that each unit of tLSTM is accomplished to gain information from its children Combined tLSTM with structure attention mechanism Evaluation: tenfold CV	a, f, m, n and o	Number of layers 1/2 Embedding dimensions 200 Hidden dimensions 300/400/500 Batch size 10/16/20 Number of epochs 30/40/50 Dropout rate 0.5/0:1 LR 0:001/0:015 LR decay 0:05 ADAM and SGD optimizer	P 88.9 R 89.3 F-1 89.1 (f)

SSD Species Specific Dataset, P Precision, R Recall, $F-1$ F-measure

Table 3 Intuition behind some popular manually crafted features used by cited papers under Strategy A

S. No	Features	Perception behind chosen features
1	AC	A protein sequence is treated as a set of signals which is then transformed in digitized form using suitable physico-chemical properties which are promoted to scrutinize protein features
2	CT	<i>k-mer</i> based assembly algorithm that divides three successively occurred nearby amino acids into one collective entity and computes the frequency of every combination in the whole sequence
3	LD	Extract fine information of protein interaction from the segments of continuous as well as discontinuous amino acids simultaneously
4	MCD	Employed the interfaces between serially remote but spatially near residues of amino acid to appropriately cover many overlying continuous and discontinuous segments present in sequence
5	Protein Signature	Signature generation approach which considers the amino acid sequence and its length and generate a numerical representation for each protein sequence

different network architectures: one by connecting the two inputs in a solo network; another using two networks for each protein separately. The evaluation of the predictor did after setting the best hyperparameters for the network and compared the obtained results with existing approaches. The training time of DeepPPI is better than SVM, AdaBoost, and RF. Further, in this trend, Wang et al. [48] predicted the PPIs by inputting a protein feature vector, which is a combination of the proposed MOS descriptor with AA classification, into a DNN. Unlike previous protein representor like AC, CT, LD, the proposed MOS descriptor has a characteristic to consider the order relationship of the whole AA sequence. The author gave suitable reasons for opting the network parameters for the task like ReLU AF, ADAM optimizer, and cross-entropy as cost function. The other parameters like network depth and width and the LR were computed for the particular method by varying their range and selected the best ones. And finally, the author trained the DNN model with AC, CT, and LD separately and compared their performance with the proposed DNN-MOS model on the benchmark dataset as well as the non-redundant dataset. Subsequently, Guo et al. presented a DL framework based on the properties of AA that contribute to the PPI information [49]. First, a feature vector was created according to the proposed descriptor named conjoint AAindex modules (CAM) which basically encodes a conjoint AA unit of protein sequence according to the AAindex database and repeating the same process for the whole protein sequence to generate a sequence profile. To scrutinize the CAM patterns from the sequence profile, multiple dense operators were employed, and then ReLU function is activated to introduce non-linearity. Finally, the LSTM layer was stacked to leverage the advantage of holding the long-term order dependencies and applied logistic regression to compute the results.

Following the same fashion of introducing the novel feature generation, Yao et al. [50] combined the DL with representation learning (RL) [51] to predict PPI. The purpose to include RL was to learn the data pattern automatically from the raw data, the resultant informative representation

then utilized by the considered DL model. The author proposed a DeepFE-PPI framework that basically utilizes the benefits of RL to represent the informative representation using Res2vec (inspired by word2vec) and benefits of DL by extracting effective features using the hierarchical multi-layer architecture and classify the PPI task. DeepFE-PPI used two separate DNN modules to squeeze out latent features from two embedding vectors and a joint module for PPI classification task via softmax function. Like Wang et al. [48], the author also selected the best-suited hyperparameters of the DL model for PPI prediction by analyzing the range of protein length, residue dimension, network depth, and protein length. Along with the standard performance measures; the author also compared the training time with different existing algorithms using the most optimized network parameters and concluded that the DeepFE-PPI holds the fourth position among SVM, DT, RF, NB, KNN, logistic regression and though the fastest algorithm is NB, their results are comparatively poor.

Inspired by the working and advancements of DNNs as well as the characteristics of different feature extraction methods, Zhang et al. introduced EnsDNN, an ensemble DNN-based approach for PPI prediction [52]. In EnsDNN, three different feature set is generated based on AC, LD, and MCD which are then fed to nine independent DNNs having different parameter settings. After training on each feature set, the resultant of 27 DNNs are combined to transform it to the final two-layer NN for the prediction. This strong and capable ensemble predictor leveraged the advantages of key information about interaction generated by three different feature extraction approaches and an assortment of 27 DNNs. To maintain the diversity, the author used different configurations of DNNs and remarked the ensemble size as 27 according to the favorable performance obtained. The model attained remarkable performance when evaluated on training datasets as well as independent datasets. Alakus et al. in 2019 proposed an LSTM architecture to resolve the common issue that occurred in PPI prediction tasks such as Operational time, low prediction accuracy, and cost [53].

Two different feature representation methods were used: Protein signature [54] and ProtVec [55]. In the protein signature method, every protein sequence is decomposed into three letters groups which are termed monomer units. For example, an AA sequence of six letters will have 4 monomer units. These monomer units are called signatures and each one has a root and two neighbors which will be arranged alphabetically and then the resultant signature will be the addition of all the obtained signatures. The ProtVec method is based on the protein-splitting process and physicochemical properties [55], the author did not fully describe this process. Once the training data get converted to their numerical form using the mentioned method, it was then fed to the LSTM architecture for further processing. The model comprised of four 1D convolutional layers followed by an average pooling layer with each, one LSTM, and one FC layer with Softmax layer for classification. Though the proposed LSTM model behaved well with both the methods but still lacks in accuracy when compared with existing approaches. Also, the author failed to prove what issues he had committed to resolving.

In a publication of 2019 [56], CNN used to deeply extract hidden features from a matrix-based biological information of protein generated by Position-Specific Scoring Matrix (PSSM). Then, prediction task was accomplished by proposing a Feature-Selective Rotation Forest algorithm (FSRF) whose main purpose is to reduce data dimension and noisy information for improving the prediction accuracy and speed up the classifier. The proposed approach was experimented of k and r dataset and then compared the result by switching the classifier to SVM and achieved the favorable outcomes from the proposed FSRF.

In the very next year, Gui et al. [57] constructed a DNN model with the intention to optimize the prediction performance using a dropout technique and used AC, CT and LD in combine. The authors performed several experiments with different dropout rates to select the appropriate one. The results proved that the inclusion of dropout to avoid overfitting helps in enhancing the performance.

In the very next year, a notable work toward the improvement of the factors that greatly affect the PPI prediction was published by Yang et al. [58]. The author proposed feature extraction and fusion method in which each AA sequence is first converted into the digitized form using physicochemical properties and then applied DWT and CWT with 25-scale mexh wavelet function so as to cover the maximum possible interaction information. Additionally, the author changes the way of inputting the protein features into the network by adopting a 'Y-type' NN model, comprising a weight-sharing Bi-RNN layer, a buffer layer, and a dense layer. The purpose of the weight-sharing scheme is to reduce the count of parameters to speed up the training using the same values of the parameters in the respective location on both sides of

the Bi-RNN layer. Additionally, a fair comparison of training time was also presented and observed the difference of 70 s (from Du's approach [47]) and 251 s (by DNN without weight-sharing scheme); thereby proved a superior model.

Another interesting and different work implemented by Jha and Saha [59] using LSTM-based classifier that integrated the features generated by two different modalities of protein *i.e.* sequence-based and structure-based information. In this approach, firstly, three types of protein representation based on three different attributes were obtained respectively from the structural representation of the proteins, and using a ResNet50 model, a corresponding feature sets were obtained. Secondly, for sequence-based information, a stacked AE was employed to generate compact feature vectors based on AC and CT. Finally, all obtained feature sets were concatenated and fed as an input to the LSTM-classifier. The objective was to improve the prediction capability and robustness of the existing methods and learn more useful information about the interaction by utilizing two protein modalities in one go. The author evaluated the prediction performance and showed the results of every possible combination of the feature sets like structural features with AC, structural features with CT, structural features with both AC and CT on the benchmark dataset.

Hanggara stated that PPI can be utilized as proof of the adequacy of herbal medication; a DNN-based approach was implemented for PPI prediction [60]. The numerical representation of protein sequence was done using CT and then used two different methods for classification: SAE and multi-layer-ELM-AE. The models are trained and evaluated with a fivefold CV and compared with each other. However, a proper explanation of any concept and details about the work were not provided.

In the very next year, a notable work in sequence-based PPI prediction was proposed by employing a hybrid classifier approach along with the combination of three feature extraction methods. The author in [61] extracted the raw features from the protein sequences using AAC, LD, and CT, which were then fused and fed to the DNN to filter out noiseless and non-redundant features, this robust and more relevant feature set were then inputted to the extreme gradient boost (XGB) classifier for the identification of PPI class. The end-to-end tree boosting XGB classifier is popularly known for its accurate and fast performance [62]. This proposed hybrid model was then evaluated on both interspecies and interspecies datasets with fivefold CV with standard performance measures and compared the results to prove the enhanced outcomes having enriched features in terms of t-statistics [63] also.

Different from usual features (AC, CT, LD) used in the PPI prediction task, Jha et al. used an amalgamation of different features for the very first time [64] and employed SAE for the PPI prediction which is ordinarily used for feature

compression. The feature vector used by SAE included the 43 features generated by three different methods: 22 Evolutionary features based on generation of a PSSM using PSI-Blast algorithm [65]; 17 structural features generated via a DL model SPIDER2 [66, 67]; 7 features generated by popularly used physicochemical properties. Some loopholes are noticed in this: SAE a generally used for removing the noise and redundant data; though the author also mentioned the same, how SAE worked as a classifier in their work was not explained anywhere; the comparison of proposed work was not satisfactory as there is enough work have been done in this area, the proposed work was compared by only one approach. Following the same trend, an ensemble of two AEs (one for interacting pairs and the second for non-interacting pairs) was used as a binary supervised classifier termed AutoPPI to predict the PPI class [68]. The feature vectors used were AC and CT. For these AEs, three types of NN architectures were used: Joint-Joint architecture which takes the features of a proteins pair as input and correspondingly returns the renovated features at the output; Siamese-Joint architecture having a shared structure at the encoder side which compresses the two proteins in a pair in two encodings and decoder works the same as previous architecture.; Siamese–Siamese architecture in which a common representation is generated by element-wise multiplication two encodings for each protein in a pair at the encoder side and the reconstruction of proteins is obtained using a shared decoder. In all three architectures, the Selu AF and Adam optimizer were used. Another notable research in this domain was proposed by Xu et al. called GRNN-PPI to predict sequence-based PPIs [69]. GRNN-PPI utilized and combined two feature extraction methods: AC and second one is a novel approach to cover evolutionary features using a proposed Mutation Spectral Radius motivated by Yu's [70] approach. Then, PCA was used to eliminate noise and redundant data from the obtained fused feature set. Lastly, for the classification purpose, a memory-based learning algorithm named General regression neural network (GRNN) [71] was used having 4 layers: input, pattern, summation, and output layer. GRNN-PPI performed well when evaluated on three benchmarks and six independent datasets and two PPI networks as well.

Other than existing numerical mapping approaches like physicochemical, character, and signal-based, an algorithm-based protein numerical mapping process was proposed for the first time by Alakus in 2021 to predict PPIs and applied on COVID-19 using DNs [72]. The author did efforts in dataset set up because of the scarcity of suitable data due to the new disease. Also according to the author, this algorithm-based mapping is the first approach in this field. This proposed algorithmic approach made use of the AVL tree because of its fast search processing and balancing properties. To generate an AVL tree, first, the one-letter code

of each AA was considered and arranged in alphabetical order and by following the insertion and deletion rules of a balanced AVL tree, the final structure was obtained. Then, the depth value of each AA was determined and converted to every AA sequence accordingly in its numerical form. Because the author compared the proposed mapping method with the other existing ones, the input sequences were mapped accordingly using every mapping approach which then underwent a normalization process. The obtained result was then fed to a DeepBiRNN for the classification. The structure of considered DeePBIRNN was: first-three layers are BiRNN with ReLU AF and the number of units were 64,32,16 respectively; followed by Flatten, Batch normalization and Dropout function; next two FC layer. The resultant performance was favorable with this novel algorithmic mapping process.

A notable experiment done for improving the performance of CNN model in PPI tasks by proposing an encoding technique [73]. The proposed Sequence-Statistics-Content is basically three-channel format method which is able to present more refined features and decrease the effect from local sequence similarity. The output of SSC, the statistical information and bigram encoding information of protein sequence, were then fed to the 2D CNN using 2D convolutional kernels that offer ample features instead of the distinct features of one hot encoding. The author then evaluated the performance using different datasets and compared the results with existing approaches. Additionally, the effect of different SSC channel combination were also shown by the author. The overall results provide a valuable insights for DN in PPI prediction task.

Figure 8 presents the best performance in terms of accuracy with the most suitable parameter settings of the various aforementioned DN approaches to predict PPIs. The performance measures by some papers [72] are either multiple or unclear, therefore, those approaches are not considered in the figure. It can be observed that approaches by [58] and [69] are performing well using Benchmark dataset and *H. pylori* dataset.

Strategy-B: Auto-Feature Engineering based PPI Prediction Approaches

To our knowledge, the first research on sequence-based PPI prediction using DNs that solely based on auto-feature engineering *i.e.* without the inclusion of manually extracted features was presented by Li et al. in the year 2018 termed as DNN-PPI [74]. For the NN architecture to learn the data, the input should be in numeral form. Therefore, the author assigned each AA a natural number randomly and accordingly converted the protein sequence. Within the proposed framework, the embedding layer captured the information regarding semantic association among AA, position-based

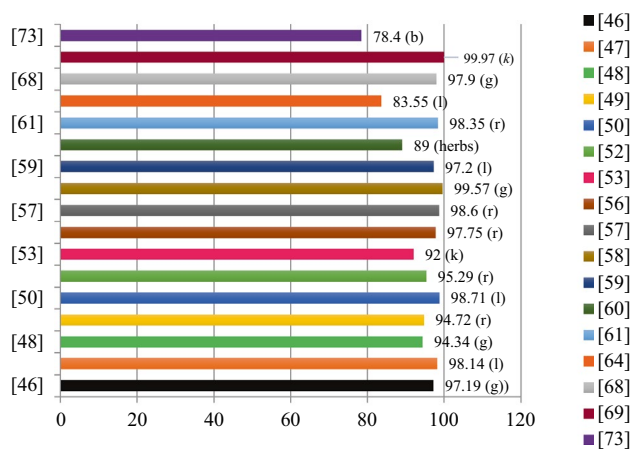


Fig. 8 Performance analysis of highest accuracy reported by various approaches of Strategy-A (in %). The dataset name is mentioned in bracket alongwith the accuracy (best). Approach used by [69] is performing best using 'k' dataset

features of protein sequences were bagged by three-layered CNNs, and short as well as long-term dependencies were covered by the LSTM layer and then the concatenated features were then fed to the FC layer with dropout to identify potential features. Besides the favorable results of DNN-PPI, the author also tested the performance by changing the number of CNN layers to 1 and 2 and concluded with no significant difference in terms of accuracy but had speedy convergence in loss with the higher number of layers. Further, Gonzalez-Lopez et al. [75] performed PPIs prediction through embedding systems and RNNs and bypass the need of feature engineering. The tokenization process was used to represent the sequence into numerical form by assigning a token (an integer) to every triplet in the sequence. In the NN, each protein's representation of the pair was fed and processed separately in two branches having similar architecture. The embedding, recurrent, and FC layers used in the architecture performed their specific roles. Along with this, two important parameters Dropout and Branch normalization were also used to avoid over-fitting and input standardization. Moreover, the schemes like early stopping and Reduce LR when stagnation was also considered to avoid wasting resources and to achieve better local minima. The observation from the results obtained by evaluation with different datasets is that the performance of the proposed Deep-SequencePPI approach is similar to other existing methods which were using hand-crafted features with DL approach and thereby concluded that if sufficient data is available, then DNs could properly model PPI prediction task without the inclusion of manually created features.

To handle huge training data with effectively capture the potential features of protein pairs, a remarkable DL approach (DPPI) was implemented by Hashemifar et al. [76] having

the generalization characteristics to be easily used for different applications with slightly tuning the parameters. The successful execution of three main modules is contributed to the design of the DPPI model. The first and core module is the Convolutional module consists of a set of filters (convolutional layer, ReLU, batch normalization, and pooling layer) responsible for mapping the protein sequences to the representation suitable for further processing by detecting pattern that characterizes the interaction information. The input in DPPI was taken as the sequence profiles, which was generated on the basis of probability using the PSI-BLAST algorithm. The next module is Random Projection (RP) consists of two FC sub-networks and is responsible to project the convoluted representation of two proteins to two different spaces. The word 'random' is used for taking the random weights so that model could learn motifs with different patterns. The outcome of the RP module is the refined representation of the proteins which are then taken as the input by the last module: The Prediction Module. The Prediction module computes the probability score by performing the element-wise multiplication on the representation taken from the previous module which indicates the interaction probability of two proteins in a pair. This Siamese-like convolutional NN behaved very well when evaluated with different benchmark datasets. The author committed that DPPI can serve as a principle model for sequence-based PPIs prediction and is generalizable to diverse applications.

Another effective approach PIPR [77] to capture the mutual influence of the protein pairs in PPI prediction was implemented by Chen et al. based on Siamese architecture. Besides binary prediction, PIPR was designed to address two more challenging tasks: estimation of binding affinity and prediction of interaction type. PIPR incorporates a deep Siamese environment of residual RCNN-based protein sequence encoder to better apprehend the potential features for PPI representation. This deep encoder was comprised of many occurrences of convolution layers with pooling and bidirectional residual gated recurrent units so as to ease the training and greatly diminish the updates of the parameters. For the numerical representation of the protein sequences, PIPR transformed the recognized AAs based on their similarity in terms of their co-occurrences as well as their electrostatic and hydrophobic properties and pre-trained the obtained embedding. The resultant AA embedding was then fed to the encoder to capture the latent information of the proteins in a pair. The output of the encoder is a refined embedding to two sequences which are then merged to generate a pair vector and passed to an MLP with Leaky ReLU [78] for PPI classification. The whole learning tasks were optimized by mean-squared loss for the estimation task of binding affinity and Cross-entropy loss for the remaining two tasks. PIPR proved promising results with effectively covered the mutual influence among the protein in a pair and ascertained the

generalization with the satisfactory results in all three challenging tasks without the inclusion of hand-crafted features.

Richoux et al. designed and compared two DL models: a FC model and a recurrent model intended to show the downsides which are needed to avoid while predicting PPIs [79]. For the numerical representation of protein sequence, a sequence vector of 24 Boolean values was considered and used one-hot encoding *i.e.* each AA is characterized by its true value at a specific position. 24 Boolean values contains: 20 usual AA, 4 other categories of AA including unknown acid also. In a FC model, the representation of two proteins were separately inserted and passed through the flatten layer. Then, the results were fed to the two FC layers with 20 units followed by batch normalization for speedy training time and to avoid over-fitting. The outputs of both the branches were then concatenated and inputted to the final FC layer having 1 unit with sigmoid function for PPI classification. The second carefully designed architecture inputted two protein vector representations to a three 1D-layered architecture having convolution, pooling, and batch normalization ended with an LSTM layer. This is clear that through all these layers, a variety of features were extracted such as local, global, spatial, and temporal features from the sequences. After feature extraction, the obtained information was then passed to the two FC layer for the classification. The author faced the time-consuming issue when tried to replace a sparse one-hot encoding with an embedding layer and achieved minor improvement in accuracy. This was also observed that dataset setup and DL model design require a lot of attention to evade DL workflow misuse.

Further, a novel algorithm-based approach was proposed based on the residual network termed ResPPI [80] comprised of residual units which are capable of full utilization of GPU for efficient computing and can extract deep features of the protein. In the proposed ResPPI algorithm, the embedding method, which is generally used for word representation in NLP task [81], is used for vector representation of AA sequences. The obtained two vectors—one for each AA sequence then concatenated and pass to the residual network (named as ResNet) to capture deep features. ResNet is designed for PPI prediction from the inspirational success of ResNet [82] in other applications. So the ResPPI algorithm is a combinational process of five residual units and each residual unit comprises of: three 2D convolution layers each followed by batch normalization and then a mapping function and ReLU; an additional Convolution layer is also present as a shortcut that connects the input features directly to the mapping function in some special case. The output after all residual units is passed to the FC layer having a softmax function for binary classification. The model was evaluated on two different datasets with six standard performance measures and then compared with other baseline methods such as

RNN, LSTM, GRU, DCNN, and SVM and the obtained performances were favorable in terms of accuracy and speed.

Apart from improving the prediction accuracy, a research work by Sledzieski [83] intended to address the limitation of training data size as well as improving generalization across species. D-SCRIPT (Deep Sequence Contact Residue Interaction Prediction Transfer), a DL method was proposed with a hypothesis that if a model, that is to be trained using sequential data, have favorable input features of protein that strongly characterizes the interaction information and well-designed model structure; can be able to generate a representation that depicts the behavior of structural interaction. D-SCRIPT model design is very similar to PIPR [76] and DPPI [77] with the inclusion of impression of protein structure. First, using the concept of Bepler and Berger's pre-trained model [84], protein embedding was constructed that included some structural information along with sequential information about each protein. The dimension of the obtained representation were then reduced in Projection module and outputs an abstract representation of protein features. For the interaction prediction, the author presented a different approach by taking a small sub-sequence and cross-checking its compatibility score in both protein sequences. This step is followed by a contact module responsible to evaluate a sparse contact map according to the obtained compatibility score. And lastly, in the interaction module, modified max-pooling operation is performed on the resultant contact map for identifying interaction probability. The performance of D-SCRIPT showed enhancement in terms of generalization and aiming to consider structural characteristics of interaction over the occurrence of protein as an interaction partner.

Hu et al. in 2022 [85] proposed a DL architecture Deep-Trio which provide an instinctual visualization for interpretable model which was an improvement over that of designed by [77]. The architecture was basically comprised of numerous convolution filters arranged in parallel fashion to extract deeper and refined protein features from the profiles. Additionally, this method considered the issue of weight polarization by employing single-protein class and masking operation and further proved its effectiveness by performing several experiments. The favorable outcomes proved the model's capability to provide an intuitive description of the inner mechanism of pairwise-input NN and demonstrate the influence of each AA residue on PPI.

The best performance analysis (in terms of accuracy) of various approaches under this section is presented by Fig. 9 with most favorable network conditions. The performance measures by some papers [79, 83] are either multiple or unclear, therefore, those approaches are not considered in the figure. The DN approach by [74] proved better and advocated the capability of auto-feature engineering.

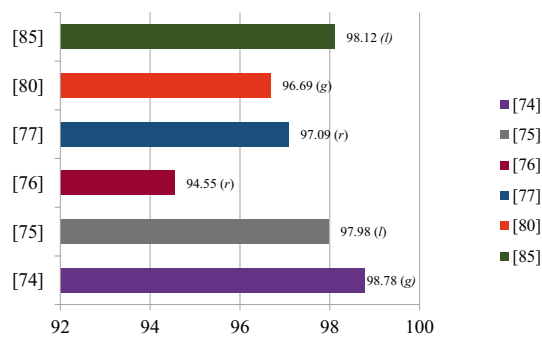


Fig. 9 Performance analysis of highest accuracy reported by various approaches of Strategy-B (in %). The dataset name is mentioned in bracket along with the accuracy (best). The best accuracy is achieved by the approach used in [74] on 'g' advocated the proficiency of auto-feature engineering

Some authors have removed sequence similarities between the training pair of proteins and testing pair of proteins for finding accurate results. The most common redundancy removal technique used is CD-HIT program [86]. The CD-HIT program is fast and greedy incremental clustering algorithm designed for larger databases. This follows a short word filtering process, which grouped proteins under certain similarity threshold (sequence identity). Among the cited papers, [47, 61, 74, 75, 83, 85] considered the same technique for the exclusion of redundancy have a sequence identity of 40%, [73] avoided the protein sequence with similarity greater than 60% and [77] varied the similarity threshold with 40, 25, 10 and 1%. The author in [56, 72] used BLAST algorithm which does pairwise comparison for finding sequence similarity [87].

Strategy-C: Prediction Using Biomedical Text Dataset

The first implementation in this category is by Hsieh et al. [88]. The author implemented the PPI identification task using a bi-directional RNN with an LSTM approach. The method includes three layers in the scenario: embedding layer which takes the protein entities in sentence form and each of its words is converted to the corresponding embedding which forms a low-dimensional vector containing real-values. Basically, this layer bagged the syntactic and semantic information by taking the effects of neighboring words. The obtained vector representation is then fed to the recurrent layer, more specifically a Bi-RNN. The resultant contextual and more refined information obtained by Bi-RNN are then taken by a FC layer for PPI classification. The author adopted two testing methods tenfold CV and cross-corpus (CC) to evaluate the performance using the two largest PPI corpora: *a* and *c* and concluded with favorable results in the CV that DNs are more suitable for extracting rich context

information from larger datasets rather than manual feature engineering.

In the very next year, a remarkable work in this domain was published by Yadav et al. [89] by utilizing dependency relationships among the names of the proteins and exploring salient features that can prove effective for the characterization of protein pairs. The major objective was to bagged-in all the key entities and relevant information from a sentence and bypass the not very important attributes so that to circumvent the limitation of existing methods and to enhance the performance. For this, a Shortest Dependency Path (SDP) was created to interpret more relevant information using a Bi-directional LSTM (Bi-LSTM). For SDP creation, a graph is developed for every sentence where nodes signify the words and edges represent the dependency relationship among the nodes obtained by Enju parser [90] and then the BFS algorithm is followed to compute the shortest distance among the protein pairs. In this way, the words that occur in the final SDP will process further rather than a complete sentence and thus created SDP embedding. Additionally, with the intention to design a generalizable and adaptable model, more salient features were explored such as Part-of-Speech (POS) and Position features with the help of Genia Tagger [91] and AE. Then, an embedding layer is used in which the embeddings of SDP, POS, and position are concatenated to generate a vector representation suitable for the Bi-LSTM as input. Further, Bi-LSTM comprises of three layers: Sequence, Max-Pooling, and MLP layer which are responsible for eliminating noise and capture contextual and maximum possible feature-rich information from the obtained embedding and make the PPIs prediction accordingly. The model was evaluated on two popular corpora and concluded with favorable results.

The same group of authors [92] implemented the same task with slight modifications in the model. They include an attention layer and used a stacking strategy in the Bi-LSTM unit. The remaining work and architecture are same as [89]. The LSTM model with multiple hidden layers having numerous memory units is termed as stacked LSTM. The author employed the vertical stacked LSTM to capture a high-level abstract demonstration of every word in the sentence. The output of this layer is the hidden state representation of its last layer which are then taken as inputs to the attention layer. The goal of the attention layer is to generate the clues that can be a deciding factor of interaction information or in a more simple words, it tells that how much attention is to be given to a particular word at the present state. It is computed by multiplying some attention weights to the obtained hidden representation. The model was evaluated on five benchmark corpora and concluded with a significant improvement over [89].

Besides basic LSTM that can only be used for investigating sequential information, tree LSTM (tLSTM) [93] can

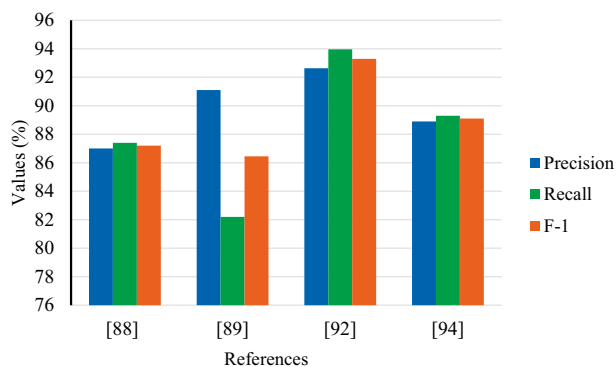


Fig. 10 Analysis of highest performance reported by cited papers under Strategy-C (in %). The attention layer approach used in [92] performed best using corpora ‘a’

be a better option for scrutinizing extra information. Ahmed et al. [94] established his PPI identification work on tLSTM and traversed the PPI-related sentences through the network topology of tree-like structure in such a way that each unit of tLSTM is accomplished to gain information from its children. Additionally, to build the final model, the author fused the output vector obtained from tLSTM to an attention mechanism to calculate the strength of attention at each unit. This fusion of tLSTM with structure attention mechanism was evaluated on five PPI corpora including large and small corpora and outperformed the traditional comparative approaches. It was also observed that due to different distribution, fewer syntactic dependencies were captured, and thereby the model with attention mechanism was performing poorly than the model without attention scheme.

Figure 10 depicts the analysis of best performance achieved by various approaches mentioned under this strategy. The details of these measures are mentioned in the Table 2. It can be clearly observed from the figure that the inclusion stacking strategy and attention layer in [92] greatly enhanced the performance using *a* copora and also proved superior to the other competitive approaches.

Figure 11 presents the count of papers published using particular strategy. It can be witnessed that although DNs are known for their auto-feature engineering capability but still there are a lot more to discover because numerous researchers are taking the help of hand-crafted features with DNs for improving the performance.

Implementation of Cited Papers

This section presents the implementation results of two papers among the cited papers. One paper is taken from Strategy-A [61] that employed a hybrid classifier (DNN-XGB) approach along with the combination of three feature extraction methods namely AAC, CT and LD. The

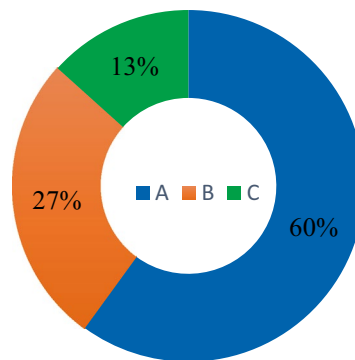


Fig. 11 Categorization of number of published papers according to Strategy

implementation was done on two datasets *k* and *r*. For this, all three features were extracted separately for each datasets. Then, two files were generated for combined positive features and combined negative features of AAC, CT and LD. Lastly, these two feature files were used by the hybrid classifier for the prediction result. The implementation result are as shown in the Fig. 12. This work was implemented on environment of 8 GB RAM and x64-based processor using MATLAB R2016a [95] software for feature generation and keras [96] library of Python 3.8.2 was used for classification.

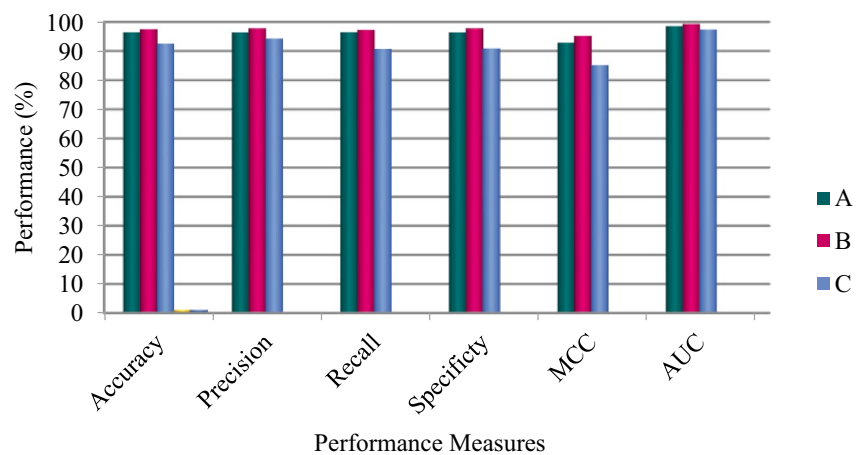
Second paper is taken from Strategy-B [75] that advocated the auto-feature engineering for PPI prediction. The implantation was done on *r* dataset using Google Colaboratory [97] environment enforcing keras library of Python 3.8. The fasta file [98] of AA sequence in taken online for tokenization and generation of n-gram dictionary. The obtained results are as shown in the Fig. 12.

The details of performance measures are mentioned in the cited papers. The observations from the Fig. 12 are that although DL architectures are known for their auto-feature engineering capability but still there are a lot more to discover because numerous researchers are taking the help of hand-crafted features with DL for improving the performance like in [61]. If the nature of DL architectures is deeply studied, like the authors in [75] did, and applied according to the problem taken then the need and effort of generating protein feature can be easily bypassed.

Comparison with State-of-the-art Methods

For better understandability of the enriched improved performance of PPI prediction using DNs, a comparison of some discussed approaches are made in this section with the state-of-the-art methods proposed for the same. Table 4 shows the best-reported results of various existing approaches suggested for the sequence-based PPI prediction in which the author used AC [13], ACC

Fig. 12 Performance analysis of manual implementation of approaches employed by [61, 75]. A: Implementation of [61] on k dataset; B: Implementation of [61] on r dataset; C: Implementation of [75] on r dataset



A: Implementation of [61] on k dataset; B: Implementation of [61] on r dataset; C: Implementation of [75] on r dataset.

Table 4 Comparison of the deliberated approaches with state-of-the-art methods

References	Approach	Acc (%)
[46]	AC + SAE^a	97.19
[52]	AC + LD + MCD^a	95.29
[57]	AC + CT + LD^a	98.6
[61]	AAC + CT + LD^a	98.35
[13]	AC + SVM	87.36
[13]	ACC + SVM	89.33
[11]	LD + SVM	88.56
[15]	MCD + SVM	91.36
[10]	CT + SVM	83.9
[99]	AC + CT + LD + MAC + E-ELM	87.5
[14]	MLD + RF	88.3
[12]	LD + KNN	86.15
[100]	Phylogenetic bootstrap	75.8
[101]	HKNN	84
[54]	Signature products	83.4
[102]	Ensemble of HKNN	86.6

^aPerformance highlighted in bold are the various approaches discussed in previous sections that used DNs for PPI prediction

[13], CT [10], LD [11], MCD [15], MLD [14] and their combinations [99] with different ML-based classifiers. Some exciting approaches like phylogenetic bootstrap [100], hyperplane distance nearest neighbor algorithm (HKNN) [101], ensemble of HKNN [102], K-local signature products [54] were also proposed. This can be clearly observed from Table 4 that the DNs are now a well-suited selection for the problem taken with favorable outcomes.

Conclusion

Recently, DL technology has come into the limelight with numerous scientific researches and has also become a hot topic in business applications. In the area of bioinformatics, where incredible advances have been made with ML, promising and more significant outcomes are expected by DL. This paper provides a comprehensive review of three architectures of DL: DNNs, CNNs and RNNs including its variants in the domain of PPI prediction using sequence information and broadly discussed the various approaches in terms of input data, objectives, and structure of the DL architecture along with their best-suited parameters.

It is observed that all considered architectures are capable to provide effective results in the considered area but to fully utilize of competencies of these approaches; there still remain several budding challenges like inadequate data, opting for the suitable architecture with favorable hyperparameters, and many more. Also, advanced and deep study is essential to scale up the popularity of DL approaches. Therefore, the detailed discussion presented herein with carefully mined every possible information can help the researchers to further explore the success in this area. It is believed that this literature survey will bring a treasured vision to assist the scholars in the applications of DNs in PPI prediction in imminent research.

Author contributions All the authors contributed equally to this research paper.

Funding Not applicable.

Availability of data and material Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval This article work have tested on the already available data in research community.

References

- Damodaran S. Amino acids, peptides and proteins. *Fennema's Food Chem.* 2008;4:425–39.
- Keskin O, Gursoy A, Ma B, Nussinov R. Principles of protein–protein interactions: what are the preferred ways for proteins to interact? *Chem Rev.* 2008;108(4):1225–44.
- Skrabanek L, Saini HK, Bader GD, Enright AJ. Computational prediction of protein–protein interactions. *Mol Biotechnol.* 2008;38(1):1–17.
- Puig O, Caspary F, Rigaut G, Rutz B, Bouveret E, Bragado- Nilsson E, et al. The tandem affinity purification (TAP) method: a general procedure of protein complex purification. *Methods.* 2001;24(3):218–29.
- Zhu H, Bilgin M, Bangham R, Hall D, Casamayor A, Bertone P, et al. Global analysis of protein activities using proteome chips. *Science.* 2001;293(5537):2101–5.
- Browne F, Zheng H, Wang H, Azuaje F. From experimental approaches to computational techniques: a review on the prediction of protein–protein interactions. *Adv Artif Intell.* 2010. <https://doi.org/10.1155/2010/924529>.
- Zhang M, Su Q, Lu Y, Zhao M, Niu B. Application of machine learning approaches for protein–protein interactions prediction. *Med Chem.* 2017;13(6):506–14.
- Sarkar D, Saha S. Machine-learning techniques for the prediction of protein–protein interactions. *J Biosci.* 2019;44(4):1–12.
- Hamp T, Rost B. Evolutionary profiles improve protein–protein interaction prediction from sequence. *Bioinformatics.* 2015;31(12):1945–50.
- Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, et al. Predicting protein–protein interactions based only on sequences information. *Proc Natl Acad Sci USA.* 2007;104(11):4337–41.
- Zhou YZ, Gao Y, Zheng YY. Prediction of protein–protein interactions using local description of amino acid sequence. In: Zhou M, Tan H, editors. *Advances in computer science and education applications*. Berlin: Springer; 2011. p. 254–62.
- Yang L, Xia JF, Gui J. Prediction of protein–protein interactions from protein sequence using local descriptors. *Protein Pept Lett.* 2010;17(9):1085–90.
- Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Res.* 2008;36(9):3025–30.
- You ZH, Chan KC, Hu P. Predicting protein–protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. *PLoS One.* 2015;10(5): e0125811.
- You ZH, Zhu L, Zheng CH, Yu HJ, Deng SP, Ji Z. Prediction of protein–protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. *BMC Bioinform.* 2014;15(15):1–9.
- Qi Y, Bar-Joseph Z, Klein-Seetharaman J. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins Struct Funct Bioinform.* 2006;63(3):490–500.
- Yu B, Chen C, Zhou H, Liu B, Ma Q. GTB-PPI: predict protein–protein interactions based on L1-regularized logistic regression and gradient tree boosting. *Genomics Proteom Bioinform.* 2020;18(5):582–92.
- Wei L, Xing P, Zeng J, Chen J, Su R, Guo F. Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier. *Artif Intell Med.* 2017;83:67–74.
- Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing. IEEE; 2013. p. 6645–6649.
- Abdel-Hamid O, Mohamed AR, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. *IEEE/ACM Trans Audio Speech Lang Process.* 2014;22(10):1533–45.
- Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning. [arXiv:1506.00019](https://arxiv.org/abs/1506.00019) [Preprint]. 2015.
- Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst.* 2012;25:1097–105.
- Kuksa PP, Min MR, Dugar R, Gerstein M. High-order neural networks and kernel methods for peptide–MHC binding prediction. *Bioinform.* 2015;31(22):3600–7.
- Li Y, Shi W, Wasserman WW. Genome-wide prediction of cis-regulatory regions using supervised deep learning methods. *BMC Bioinform.* 2018;19(1):1–14.
- Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge: MIT Press; 2016.
- LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature.* 2015;521(7553):436–44.
- Vincent P, Larochelle H, Bengio Y, Manzagol PA. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*; 2008. p. 1096–1103.
- Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA, Bottou L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res.* 2010;11(12):3371–408.
- Tang B, Pan Z, Yin K, Khateeb A. Recent advances of deep learning in bioinformatics and computational biology. *Front Genet.* 2019;10:214.
- Bottou L. Stochastic gradient learning in neural networks. *Proc Neuro-Nimes.* 1991;91(8):12.
- Krogh A, Hertz JA. A simple weight decay can improve generalization. In: *Advances in neural information processing systems*. San Francisco, CA: Morgan Kaufmann; 1992. p. 950–7.
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res.* 2014;15(1):1929–58.
- Moon T, Choi H, Lee H, Song I. Rnndrop: A novel dropout for RNNs in ASR. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE; 2015. p. 65–70.
- Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR; 2015. p. 448–456.
- Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In *ICML*; 2010.
- Spencer M, Eickholt J, Jianlin C. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Trans Comput Biol Bioinform.* 2005;12(1):103–12.
- Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput.* 2006;18(7):1527–54.
- Vennerød CB, Kjærø A, Bugge ES. Long short-term memory RNN. [arXiv:2105.06756](https://arxiv.org/abs/2105.06756). [Preprint]; 2021.

39. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Networks*. 1994;5(2):157–66.
40. Nosouhian S, Nosouhian F, Khoshouei AK. A review of recurrent neural network architecture for sequence learning. Comparison between LSTM and GRU; 2021.
41. Sorin V, Barash Y, Konen E, Klang E. Deep learning for natural language processing in radiology—fundamentals and a systematic review. *J Am Coll Radiol*. 2020;17(5):639–48.
42. Dhruv P, Naskar S. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): a review. In: Swain D, Pattanaik P, Gupta P, editors. *Machine learning and information processing*. Singapore: Springer; 2020. p. 367–81.
43. Kim P. *Convolutional neural network*. MATLAB deep learning. Berkeley, CA: Apress; 2017. p. 121–47.
44. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET). IEEE; 2017. p. 1–6.
45. Zheng S, Dharssi S, Wu M, Li J, Lu Z. Text mining for drug discovery. *Methods Mol Biol*. 2019;1939:231–52.
46. Sun T, Zhou B, Lai L, Pei J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinform*. 2017;18(1):1–8.
47. Du X, Sun S, Hu C, Yao Y, Yan Y, Zhang Y. DeepPPI: boosting prediction of protein–protein interactions with deep neural networks. *J Chem Inf Model*. 2017;57(6):1499–510.
48. Wang X, Wu Y, Wang R, Wei Y, Gui Y. A novel matrix of sequence descriptors for predicting protein–protein interactions from amino acid sequences. *PLoS ONE*. 2019;14(6): e0217312.
49. Guo Y, Chen X. A deep learning framework for improving protein interaction prediction using sequence properties. *bioRxiv*, 843755; 2019
50. Yao Y, Du X, Diao Y, Zhu H. An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*. 2019;7: e7126.
51. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(8):1798–828.
52. Zhang L, Yu G, Xia D, Wang J. Protein–protein interactions prediction based on ensemble deep neural networks. *Neurocomputing*. 2019;324:10–9.
53. Alakus TB, Turkoglu I. Prediction of protein–protein interactions with LSTM deep learning model. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE; 2019. p. 1–5.
54. Martin S, Roe D, Faulon JL. Predicting protein–protein interactions using signature products. *Bioinformatics*. 2005;21(2):218–26.
55. Asgari E, Mofrad MR. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS ONE*. 2015;10(11): e0141287.
56. Wang L, Wang HF, Liu SR, Yan X, Song KJ. Predicting protein–protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. *Sci Rep*. 2019;9(1):1–12.
57. Gui YM, Wang RJ, Wang X, Wei YY. Using deep neural networks to improve the performance of protein–protein interactions prediction. *Int J Pattern Recognit Artif Intell*. 2020;34(13):2052012.
58. Yang L, Han Y, Zhang H, Li W, Dai Y. Prediction of protein–protein interactions with local weight-sharing mechanism in deep learning. *BioMed Res Int*. 2020. <https://doi.org/10.1155/2020/5072520>.
59. Jha K, Saha S. Amalgamation of 3D structure and sequence information for protein–protein interaction prediction. *Sci Rep*. 2020;10(1):1–14.
60. Hanggara FS, Anam K. Sequence-based protein–protein interaction prediction using greedy layer-wise training of deep neural networks. In *AIP Conference Proceedings*, Vol. 2278, No. 1. AIP Publishing LLC; 2020. p. 020050
61. Mahapatra S, Gupta VRR, Sahu SS, Panda G. Deep neural network and extreme gradient boosting based Hybrid classifier for improved prediction of Protein–Protein interaction. *IEEE/ACM Trans Comput Biol Bioinform*. 2021;19:155–65.
62. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*; 2016. p. 785–794
63. Wong TT. Parametric methods for comparing the performance of two classification algorithms evaluated by k-fold cross validation on multiple data sets. *Pattern Recogn*. 2017;65:97–107.
64. Jha K, Saha S, Tanveer M. Prediction of protein–protein interactions using stacked auto-encoder. *Trans Emerg Telecommun Technol*. 2021. <https://doi.org/10.1002/ett.4256>.
65. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25(17):3389–402.
66. Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, et al. Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning. *Sci Rep*. 2015;5(1):1–11.
67. Heffernan R, Dehzangi A, Lyons J, Paliwal K, Sharma A, Wang J, et al. Highly accurate sequence-based prediction of half-sphere exposures of amino acid residues in proteins. *Bioinformatics*. 2016;32(6):843–9.
68. Czibula G, Albu AI, Bocicor MI, Chira C. AutoPPI: An Ensemble of deep autoencoders for protein–protein interaction prediction. *Entropy*. 2021;23(6):643.
69. Xu H, Xu D, Zhang N, Zhang Y, Gao R. Protein–protein interaction prediction based on spectral radius and general regression neural network. *J Proteome Res*. 2021;20(3):1657–65.
70. Yu A, Lu M, Tian F. On the spectral radius of graphs. *Linear Algebra Appl*. 2004;387:41–9.
71. Specht DF. A general regression neural network. *IEEE Trans Neural Networks*. 1991;2(6):568–76.
72. Alakus TB, Turkoglu I. A novel protein mapping method for predicting the protein interactions in COVID-19 disease by deep learning. *Interdiscip Sci Comput Life Sci*. 2021;13(1):44–60.
73. Wang Y, Li Z, Zhang Y, Ma Y, Huang Q, Chen X, et al. Performance improvement for a 2D convolutional neural network by using SSC encoding on protein–protein interaction tasks. *BMC Bioinform*. 2021;22(1):1–16.
74. Li H, Gong XJ, Yu H, Zhou C. Deep neural network based predictions of protein interactions using primary sequences. *Molecules*. 2018;23(8):1923.
75. Gonzalez-Lopez F, Morales-Cordovilla JA, Villegas-Morcillo A, Gomez AM, Sanchez V. End-to-end prediction of protein–protein interaction based on embedding and recurrent neural networks. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE; 2018. p. 2344–2350
76. Hashemifar S, Neyshabur B, Khan AA, Xu J. Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*. 2018;34(17):i802–10.
77. Chen M, Ju CJT, Zhou G, Chen X, Zhang T, Chang KW, et al. Multifaceted protein–protein interaction prediction based on siamese residual RCNN. *Bioinformatics*. 2019;35(14):i305–14.

78. Maas AL et al. Rectifier nonlinearities improve neural network acoustic models. In ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Vol. 30; 2013. p. 3.
79. Richoux F, Servantie C, Borès C, Téletchéa S. Comparing two deep learning sequence-based models for protein-protein interaction prediction. [arXiv:1901.06268](https://arxiv.org/abs/1901.06268). [preprint]; 2019
80. Lu S, Hong Q, Wang B, Wang H. Efficient resnet model to predict protein-protein interactions with GPU computing. *IEEE Access*. 2020;8:127834–44.
81. Nikfarjam A, Sarker A, O'connor K, Ginn R, Gonzalez G. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *J Am Med Inform Assoc*. 2015;22(3):671–81.
82. Wu Z, Shen C, Van Den Hengel A. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recogn*. 2019;90:119–33.
83. Sledzieski S, Singh R, Cowen L, Berger B. Sequence-based prediction of protein-protein interactions: a structure-aware interpretable deep learning model. *bioRxiv*; 2021
84. Beppler T, Berger B. Learning protein sequence embeddings using information from structure. [arXiv:1902.08661](https://arxiv.org/abs/1902.08661) [preprint]; 2019
85. Hu X, Feng C, Zhou Y, Harrison A, Chen M. DeepTrio: a ternary prediction system for protein-protein interaction using mask multiple parallel convolutional neural networks. *Bioinformatics*. 2022;38(3):694–702.
86. Li W, Cd-hit GA. A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*. 2006;22(13):1658–9.
87. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol*. 1990;215(3):403–10.
88. Hsieh YL, Chang YC, Chang NW, Hsu WL. 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)*; 2017. p. 240–245
89. Yadav S, Kumar A, Ekbal A, Saha S, Bhattacharyya P. Feature assisted bi-directional LSTM model for protein-protein interaction identification from biomedical texts. [arXiv:1807.02162](https://arxiv.org/abs/1807.02162). [preprint]; 2018
90. Miyao Y, Sætre R, Sagae K, Matsuzaki T, Tsujii JI. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of ACL-08: HLT*; 2008. p. 46–54
91. Tsuruoka Y, Tateishi Y, Kim JD, Ohta T, McNaught J, Ananiadou S, Tsujii JI. Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic conference on informatics*. Springer, Berlin; 2005. p. 382–392
92. Yadav S, Ekbal A, Saha S, Kumar A, Bhattacharyya P. Feature assisted stacked attentive shortest dependency path based Bi-LSTM model for protein-protein interaction. *Knowl-Based Syst*. 2019;166:18–29.
93. Tai KS, Socher R, Manning CD. Improved semantic representations from tree-structured long short-term memory networks. [arXiv:1503.00075](https://arxiv.org/abs/1503.00075). [preprint]; 2015
94. Ahmed M, Islam J, Samee MR, Mercer RE. Identifying protein-protein interaction using tree LSTM and structured attention. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE; 2019. p. 224–231
95. Lee HH. *Programming with MATLAB 2016*. Mission: SDC Publications; 2016.
96. Gulli A, Pal S. *Deep learning with Keras*. Birmingham: Packt Publishing Ltd; 2017.
97. Carneiro T, Da Nóbrega RVM, Nepomuceno T, Bian GB, De Albuquerque VHC, Reboucas Filho PP. Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*. 2018;6:61677–85.
98. Pearson WR. Using the FASTA program to search protein and DNA sequence databases. In: *Computer analysis of sequence data*. Totowa: Humana Press; 1994. p. 307–31.
99. You ZH, Lei YK, Zhu L, Xia J, Wang B. Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. *BMC Bioinform*. 2013;14:69–75.
100. Bock JR, Gough DA. Whole-proteome interaction mining. *J Bioinform*. 2003;19:125–34.
101. Nanni L. Hyperplanes for predicting protein-protein interactions. *Neurocomputing*. 2005;69:257–63.
102. Nanni L, Lumini A. An ensemble of K-local hyperplanes for predicting protein-protein interactions. *Bioinformatics*. 2006;22:1207–10.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.