Received: 22 November 2020 | Accepted: 21 February 2021

DOI: 10.1002/ett.4256

RESEARCH ARTICLE

WILEY

Prediction of protein-protein interactions using stacked auto-encoder

Kanchan Jha[1] | **Sriparna Saha**[1] | **M. Tanveer**[2]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, Bihar, India

[2]Department of Mathematics, Indian Institute of Technology Indore, Simrol, Madhya Pradesh, India

Correspondence
Kanchan Jha, Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, Bihar 801103, India.
Email:jha.kanchan15@gmail.com

Funding information
Science of Engineering research Board India, Grant/Award Number: ECR/2017/001915

Abstract

Protein-protein interactions (PPIs) play essential roles in understanding the protein functions and the corresponding pathways which are involved in various biological processes, as well as help in understanding the cause and growth of diseases. Several computational methods such as Support Vector Machine and decision tree are popularly used along with the experimental methods to address the PPIs problem. Such algorithms consider different protein features, including protein sequence, genomes, protein structure, function, topology of the PPIs network, and those that combine multiple aspects. Nowadays, Deep learning (DL) algorithms are successfully used in solving problems in different domains. So, in this paper, we have used stacked auto-encoder as one of the DL methods in solving the problem of PPIs. This model takes the input 92-length feature vector, which is the integration of features extracted from the protein sequence using different methods. The feature vector consists of evolutionary features obtained by PSI-BLAST algorithm, predicted structural properties obtained by SPIDER2, and seven physicochemical properties of amino acids. The key novelty of the current study lies in extracting useful features to solve the PPI problem. The results obtained by our method of feature extraction are compared with those obtained by other feature extraction methods such as Autocovariance and Conjoint-triad, and our proposed feature extraction method is found to be relatively more accurate.

1 | INTRODUCTION

Protein is a vital ingredient of all living organisms. It is composed of 20 different amino acids, which are organic compounds. Protein-protein interactions (PPIs) are defined as the physical contacts established between two or more proteins resulting from biochemical events. The PPIs are the complex network of reactions that play essential roles in regulating and executing most biological processes. The core of biological functioning is fundamentally influenced by the network of PPIs and constitutes the basis for the cell's structures and functions.[1] Several high-throughput experimental methods like Yeast two-Hybrid,[2] Tandem Affinity Purification,[3] and Mass Spectrometric Protein Complex Identification[4] have been used to discover PPIs. However, their accuracy values are questionable due to the occurrence of a large number of False Positives (FPs) and False Negatives (FNs).[5] Also, experimental techniques for PPI recognition incur huge time and cost. In this case, fast and scalable machine-learning algorithms are proved to be extremely useful for predicting novel PPIs and, when used in conjunction, may improve the effectiveness of experimental recognition of PPIs.

Abbreviations: AC, autocovariance; CT, conjoint-triad; SAE, stacked auto-encoder

Machine learning approaches are adopted successfully to solve the problems in various domains. Some real-life applications of machine learning techniques include industrial informatics,[6-8] health informatics,[9] services based on the Internet of Things.[10-14]

1.1 | Literature survey

It was in the year of 2001 that machine learning techniques were first being used for the prediction of PPIs, the credit for which goes to a few research groups which undertook this task independently.[15-17] The process of predicting the occurrence of PPIs involves an input of the sequences or structures of the combining proteins and the probability of their interactions as the corresponding output. The input elements to a machine learning algorithm which are used to create statistical models for prediction purpose are generally known as "features." We can classify the machine learning algorithms used in the prediction of PPIs broadly into two main categories based on the need of labeling the input variables according to the expected outcome: supervised and unsupervised. In supervised learning, the input and the output values are given and are used to infer a mapping function. This function can be then used to predict the output for a new set of inputs. Unsupervised learning, on the other hand, takes an altogether different approach. It takes up unlabeled training data and comes out with a meaningful conclusion by unearthing the hidden structures within them. Artificial Neural Networks, Bayesian inference, Support Vector Machines, and decision tree-based methods such as Random Forest are some of the illustrations of supervised machine learning algorithms that are used for PPIs prediction.[17-19] While clustering techniques such as k-means, single-linkage, and spectral clustering are examples of unsupervised machine-learning methods which are used for PPIs prediction.[20,21] A general inference was that computational prediction of PPIs exhibited almost analogous accuracy values to those shown by large-scale experimental PPIs datasets.[22]

Of late, many computational methods have been developed to solve the PPIs problems. Some of them tried to extract new protein information while others focused on developing new machine learning algorithms. Various feature extraction methods are available in the literature to predict PPIs. Feature extraction methods can be categorized broadly into six categories based on the features of the input proteins used in the process of prediction.[23] The use of amino acid sequences of target proteins as information has been used by several methods such as autocovariance (AC)[19] and conjoint-triad (CT).[24] The ease of availability of sequence information for all proteins in an organism when its genome sequence is known provides a clear advantage of using such information. There exist a variety of methods to predict PPIs such as sequence-based, comparative genomics-based, gene co-expression based, protein tertiary structure-based, PPI network topology based, and the integration of multiple features obtained by different methods. Combining several methods helps incorporate advantages of each of them and hence results in an increase in the prediction confidence and coverage. In this work, we have integrated different features obtained by various methods. We have used evolutionary features obtained by using PSI-BLAST algorithm,[25] structural properties predicted by SPIDER2,[26,27] and seven common physicochemical properties.[28]

1.2 | Motivation and contribution

Deep learning algorithms have significant advantages over traditional machine learning methods in terms of their capacity in managing large-scale raw data, handling complex information and ability to self-learn other handy features.[29] Some of these algorithms have been able to imitate deep neural pathways and connections and also to some extent the ability of the human brain to learn processes. This has endowed them the capacity to have some of the most remarkable applications in the field of speech and image recognition,[30,31] image super resolution,[32] natural language understanding and processing,[33,34] decision-making,[35] sentiment analysis,[36] recommender system,[37] and clickbait detection.[38] In the past few years, these algorithms have been applied for solving different problems of bioinformatics and have been able to manage an enormous amount of data and their large dimensions generated by high throughput techniques.[39-43] Several significant works have been carried out to predict protein functions. Spencer et al with the help of a deep belief network predicted the secondary structure of proteins, with an accuracy of 80.7%.[44] This accuracy was further improved to 84% by Sheng et al using deep convolutional neural fields.[45] Works of Heffernan et al helped predict backbone angles and solvent accessible surface areas along with the usual prediction of the secondary structures.[26] Computational biology has seen various applications of deep learning recently, the summary of which can be found in a review article.[46]

In this work, we have used a stacked auto-encoder (SAE), one of the deep learning models generally used for feature compression. The input feature vector given to our model is the integration of several features obtained by different methods. This work's primary motivation is to illustrate the effectiveness of feature vectors, which integrate features obtained by different methods from protein sequences in a SAE framework to predict PPIs. The results that we obtained are comparable to older methods such as AC and CT.

The notable contributions of the current work are listed below:

- To the best our knowledge, the feature combination used in this work is being used for the first time to predict PPIs.
- The extracted new features from the protein sequences are further utilized in a SAE framework to predict PPIs.

This paper is further divided into three sections. Section 2 explains the working of SAE and feature extraction methods to get feature vectors from protein sequences which are input to the model. Section 3 constitutes experimental results and comparative study which is followed by conclusion in the Section 4.

2 | METHODOLOGY

This section contains a detailed description of the working of the model (SAE) to predict PPIs and different methods to extract features from protein sequences.

2.1 | Problem statement

Suppose we are given two protein sequences P1 and P2 having internal representations as

P1: MARDKLMNWEGHTREDTGCCTCATTCGA
P2: DKLCFGKLHRDLKENTGYSKSSVAMKYIERTH

where each symbol represents a single amino acid. The names of different amino acids with the corresponding symbols are presented in Table 1. The problem statement is to determine whether two protein sequences interact or not.

2.2 | SAE for classification

Auto-encoder,[47] an example of deep learning model, falls under the category of unsupervised learning techniques. There are three layers to it: input layer, hidden layer, and output layer. The process to train an auto-encoder consists of encoder function and decoder function. The mapping of input information into hidden representation is done by the encoder, whereas, the decoder deciphers the hidden representation to get the input structure. Suppose X_n represents the unlabeled input dataset, h_n stands for the hidden encoder vector evaluated from x_n, and x' represents the decoder vector of the output layer. Hence, the encoding process is given by the formula:

$$h_n = f(W_1 x_n + b_1), \tag{1}$$

where f represents the encoding function, W_1 represents the weight matrix of the encoder, and b_1 represents the bias vector.

The process of decoding can be mathematically defined as:

$$x'_n = g(W_2 h_n + b_2), \tag{2}$$

where g represents the decoding function, W_2 represents the weight matrix of the decoder, and b_1 represents the bias vector.

TABLE 1 The values of seven physicochemical properties of amino acids.(a, steric parameters; b, polarizability; c, volume; d, hydrophobicity; e, isoelectric point; f, helix probability; g, sheet probability)

Name	Symbol	a	b	c	d	e	f	g
Alanine	A	1.28	0.05	1.00	0.31	6.11	0.42	0.23
Cysteine	C	1.77	0.13	2.43	1.54	6.35	0.17	0.41
Aspartate	D	1.60	0.11	2.78	-0.77	2.95	0.25	0.20
Glutamate	E	1.56	0.15	3.78	-0.64	3.09	0.42	0.21
Phenylalanine	F	2.94	0.29	5.89	1.79	5.67	0.30	0.38
Glycine	G	0.00	0.00	0.00	0.00	6.07	0.13	0.15
Histidine	H	2.99	0.23	4.66	0.13	7.69	0.27	0.30
Isoleucine	I	4.19	0.19	4.00	1.80	6.04	0.30	0.45
Lysine	K	1.89	0.22	4.77	-0.99	9.99	0.32	0.27
Leucine	L	2.59	0.19	4.00	1.70	6.04	0.39	0.31
Methionine	M	2.35	0.22	4.43	1.23	5.71	0.38	0.32
Asparagine	N	1.60	0.13	2.95	-0.60	6.52	0.21	0.22
Proline	P	2.67	0.00	2.72	0.72	6.80	0.13	0.34
Glutamine	Q	1.56	0.18	3.95	-0.22	5.65	0.36	0.25
Arginine	R	2.34	0.29	6.13	-1.01	10.74	0.36	0.25
Serine	S	1.31	0.06	1.60	-0.04	5.70	0.20	0.28
Threonine	T	3.03	0.11	2.60	0.26	5.60	0.21	0.36
Valine	V	3.67	0.14	3.00	1.22	6.02	0.27	0.49
Tryptophan	W	3.21	0.41	8.08	2.25	5.94	0.32	0.42
Tyrosine	Y	2.94	0.30	6.47	0.96	5.66	0.25	0.41

The objective of the auto-encoder is to learn a compressed representation of input while minimizing the reconstruction error. The error function is defined as:

$$\phi(\Theta) = \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} L(x^i, x'^i), \tag{3}$$

where L is the loss function $L(x, x') = \|x - x'\|^2$.

The structure of SAE is generated by building up n auto-encoders into n latent layers which are done by a layer-wise learning algorithm which is unsupervised in nature. This is followed by fine-tuning by a supervised method. So there are three steps to the SAEs-based method:

- The first step involves training the first auto-encoder by considering the input data and obtaining the learned feature vector.
- Secondly, the feature vector of the earlier layer is considered as the input for the next layer, and this process is repeated over and over until the training process completes.
- The last step involves using the backpropagation algorithm (BP) once training of all the hidden layers is done. This ensures that the cost function is minimized and the weights are updated with labeled training set to ensure fine-tuning is achieved.

The structure of SAE with one hidden/latent layer used for PPIs classification task is shown in Figure 1.

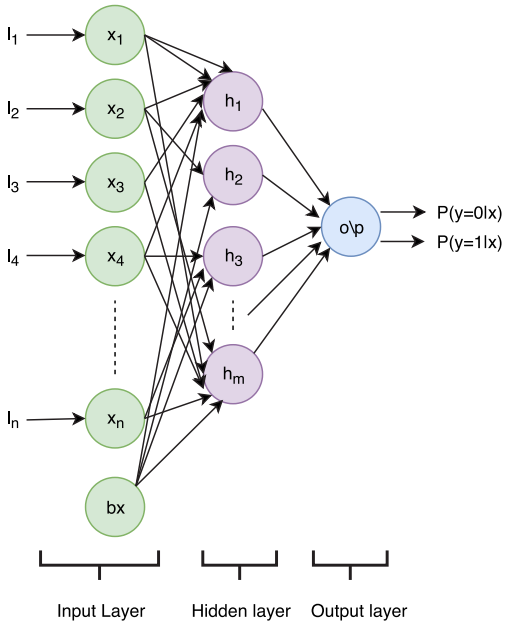

FIGURE 1 Stacked auto-encoder with 1 auto-encoder having 1 latent layer for binary classification

2.3 | Input features

To predict whether two proteins interact or not given protein sequences, several researchers have used machine learning or deep learning algorithms. The inputs to these models are feature vectors that are extracted from protein sequences. Various types of feature extraction methods are available to extract the feature vectors from protein sequences. We have extracted features using different methods and then integrated them into one feature vector. This feature vector consists of the following features.

2.3.1 | Evolutionary features

The evolutionary features consist of a total of 22 features for each amino acid in a given protein sequence. Suppose the length of the protein sequence is L. A matrix of (Lx20) is obtained through the use of a Position-Specific Scoring Matrix (PSSM), generated by three iterations of the PSI-BLAST algorithm[25] against the NCBI's Non-Redundant (NR) sequence database for each protein. The PSSM provides us a tool to measure the closeness of any sequence to the collected sequences used to create the scoring matrix. Shanon entropy is also one of the properties which is computed as it shows the amount of information in these probabilities per residue.[48] Further, a mean of the total Shannon entropy for the whole protein is used as an input feature of the general preservation of the whole protein. Altogether 22 evolutionary features are obtained for each amino acid in a given sequence.

2.3.2 | Structural features

The prediction of PPIs can be made easier and simpler if there is information regarding the 3D structure of proteins or computational modeling of the structures can be made reliable. There lies a difficulty as structures of a very few proteins are known. So, to avoid this problem, we have used secondary structures of protein pairs along with other useful information such as solvent accessible surface area of proteins, half-sphere exposure (HSE) of amino acid residues based on Cα and Cβ atoms, Contact number (CN), sine/cosine of the backbone angle values.[26,27] All these structural features of proteins are predicted by SPIDER2, which is also a deep learning-based model.

2.3.3 | Physicochemical properties

Meiler et al provided the seven generally used physicochemical properties of amino acids considered as features.[28] These are steric parameters, hydrophobicity, volume, polarizability, isoelectric point, helix probability, and sheet probability.

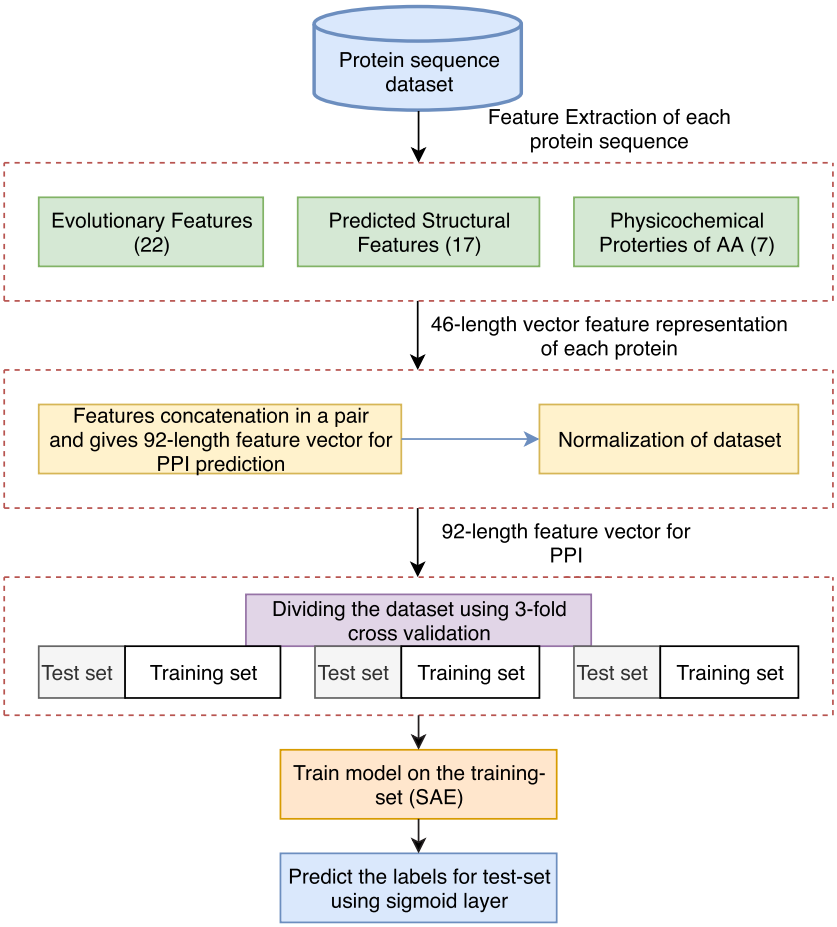

FIGURE 2 Flowchart of the proposed method for protein-protein interactions prediction

Table 1 presents the values of these properties for each amino acid of proteins. Some methods, such as AC, used these physicochemical properties to predict PPIs but not directly. The intention is to gather information regarding the periodicity of these properties along with a sequence. In the present work, gauging of the direct effect of these properties in PPIs prediction is intended and for this, these features are taken directly with some other essential features.

Thus, there are a total of 46 features we obtained using the above-mentioned methods for each amino acid in a given protein sequence. For the L length protein sequence, we obtained a feature vector of length ($L \times 46$). To make it of uniform length for every protein sequence, we have considered the mean of each feature over the entire sequence and have a feature vector of 46-length. Finally, the feature vectors of the protein pairs are concatenated and a 92-length feature vector is generated and this is given as the input to the model. The overall methodology of this work is depicted in Figure 2.

2.4 | Sequence-based features

Sun et al used the AC and CT methods to extract features separately from protein sequences.[49] These methods are explained below:

2.4.1 | AC method

One of the popular methods to derive features from protein sequences is the AC method. This method of protein-encoding explains the correlation and interaction between variables at different positions. The following equation is used to codify protein sequences:

$$\mathrm{AC}_{\mathrm{lag},n} = \frac{1}{l-\mathrm{lag}} \sum_{m=1}^{l-\mathrm{lag}} \left(X_{m,n} - \sum_{m=1}^{l} X_{m,n} \right) * \left(X_{(m+\mathrm{lag}),n} - \sum_{m=1}^{l} X_{m,n} \right), \tag{4}$$

TABLE 2 Grouping of 20 amino acids into seven clusters based on the dipoles and volume of the side chains

Group-1	Group-2	Group-3	Group-4	Group-5	Group-6	Group-7
A, G, V	C	F, I, L, P	M, S, T, Y	H, N, Q, W	K, R	D, E

where X is the protein sequence, l refers to the length of sequence X, m represents the location of amino acid in sequence X, n is the nth descriptor, and $X_{m,n}$ is the normalized nth descriptor value for mth amino acid. By using this method, the protein sequences having different lengths are converted into vectors of equal length ($n \times lag$). Sun et al. have used the value for n as seven which refers to the seven physicochemical properties[49] and the value of lag is considered to be 30.[19] This gives a vector containing 210 (7×30) numbers which represents a protein sequence.

2.4.2 | CT method

The CT[24] method to encode the protein sequences is the method that uses only their sequence information. The process of transforming protein sequence into a feature vector is divided into the following steps:

- The first step involves the clustering of 20 amino acids into seven groups depending on their dipole and side-chain volumes. Table 2 presents the group number of each amino acid.
- After that, each variable of a given sequence is replaced with its cluster/group number.
- Then, a sliding window of length 3 is used that slides from N-terminus to C-terminus of protein sequence one step at a time.
- Finally, the protein sequence is represented by a vector of length 343 ($7 \times 7 \times 7$) that contains the count of the combinations of each number. If the pattern of 3 numbers is not in the protein sequence, then that count is 0.

3 | EXPERIMENTAL RESULTS

This section contains a brief introduction of datasets (available at http://www.csbio.sjtu.edu.cn/bioinf/LR_PPI/Data.htm)[50] used in this work, performance metrics, which measure the prediction capabilities of the model, experimental results, and their thorough analysis.

3.1 | Datasets

The contents of this dataset include the positive samples (PPIs) belonging to the human protein references database (HPRD, 2007 version). All the duplicate interactions have been removed after which a total of 36 630 positive pairs are remained. Proteins from distinct subcellular locations were paired which resulted in the generation of negative samples (noninteraction pairs). Swiss-Prot database, version 57.3, was considered for information regarding the location of the protein subcellular database. The criteria used were as follows: (1) All collections were to be strictly from human proteins. (2) Dubious or unclear annotations such as "potential," "probable," "probably," "maybe," or "by similarity," if found, result in the exclusion of such sequences. (3) If more than one location is associated with a particular sequence, that sequence is deemed similar and hence excluded. (4) Exclusion of fragments was made sure by excluding all those sequences with annotation "fragment" and sequences having less than 50 amino acids over the possibility of them being fragments too. A total of 2184 different proteins from six different subcellular neighborhoods (cytoplasm, nucleus, endoplasmic reticulum, Golgi apparatus, lysosome, and mitochondrion) were obtained. A random pairing was done between two proteins belonging to different subcellular localities. This was followed by the incorporation of negative pairs from,[51] as a result, a total of 36 480 negative pairs were generated. Those pairs which had odd amino acids like U and X were rejected and amount to a total of 36 545 positive samples and 36 323 negative samples in the culminating of the benchmark dataset. Table 3 shows the examples of the PPIs dataset. It consists of proteins id's, amino acid sequences of proteins, and information about the interactions between proteins in pairs. The protein id is used to identify each protein in the protein database uniquely.

TABLE 3 Examples of dataset

S.No.	Protein Id's	Protein sequences	Interacting?
1	NP_003336.1	MSGIALSRLAQERKAWRKDHPFGFVAVPTKNPDGTMNLMNWECAIPGKKGTPWEGGLF KLRMLFKDDYPSSPPKCKFEPPLFHPNVYPSGTVCLSILEEDKDWRPAITIKQILLGIQEL LNEPNIQDPAQAEAYTIYCQNRVEYEKRVRAQAKKFAPS	No
	NP_004093.1	MVDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKN TEISFKLGVEFDETTADDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTH GTAVCTRTYEKEA	
2	NP_068739.1	MAPRPLLLLLLLLGGSAARPAPPRARRHSDGTFTSELSRLREGARLQRLLQGLVGKRSEQ DAENSMAWTRLSAGLLCPSGSNMPILQAWMPLDGTWSPWLPPGPMVSEPAGAAAEGTLRPR	Yes
	NP_002971.2	MRPHLSPPLQQLLLPVLLACAAHSTGALPRLCDVLQVLWEEQDQCLQELSREQTGDLGTE QPVPGCEGMWDNISCWPSSVPGRMVEVECPRFLRMLTSRNGSLFRNCTQDGWSETFPR PNLACGVNVNDSSNEKRHSYLLKLKVMYTVGYSSSLVMLLVALGILCAFRRLHCTRNYIH MHLFVSFILRALSNFIKDAVLFSSDDVTYCDAHRAGCKLVMVLFQYCIMANYSWLLVEGL YLHTLLAISFFSERKYLQGFVAFGWGSPAIFVALWAIARFLEDVGCWDINANASIWWIIRG PVILSILINFILFINILRILMRKLRTQETRGNEVSHYKRLARSTLLLIPLFGIHYIVFAFSPEDA MEIQLFFELALGSFQGLVVAVLYCFLNGEVQLEVQKKWQQWHLREFPLHPVASFSNSTKA SHLEQSQGTCRTSII	

TABLE 4 Characteristics of dataset

Dataset	Number of samples	Number of positive samples	Number of negative samples
Human protein-protein interactions	9484	4749	4735

TABLE 5 Different models: hyper-parameters and input-output shape

Models	Input shape	Output shape	Hyperparameters
Proposed	92	75	Activation=Sigmoid, units=75, lr=1, mom=0.5
SAE_AC[49]	420	100	Activation=Sigmoid, units=100, lr=1, mom=0.5
SAE_CT[49]	686	100	Activation=Sigmoid, units=100, lr=1, mom=0.5

Our model is based on the integration of multiple features and one of those is evolutionary feature. The evolutionary features are obtained by calculating PSSM, which is a little time-consuming task. So, because of the limited time, we extract features only for 9484 protein pairs. Out of them, the number of positive samples is 4749, and the count of negative pairs is 4735. Table 4 presents the statistics of the human PPIs dataset that we have used in our work.

3.2 | Experimental setup

To conduct this experiment, we have used SAE-based model. This model is implemented in Keras, which is a python-based framework. The learning rate and momentum are the same as used in Reference 49. All the parameters and hyper-parameters of our model and the earlier used models are mentioned in Table 5. In Table 5, *lr* is the learning rate of the model and *mom* represents momentum. The proposed approach takes integrated feature vectors (evolutionary features, predicted structural properties, physicochemical properties) as input. SAE_AC[49] is the model based on input features extracted by the AC method, whereas SAE_CT[49] is the model based on a CT method to extract features. We have tried other values for learning rate, momentum, and the number of hidden units of the model, but the changes in result are insignificant or sometimes models performed poorly.

3.3 | Evaluation criteria

PPIs prediction comes under the class of binary classification problem which generally has only two categories: the first one is the "positive" (p) category which has all the interacting proteins and the second one is the "negative" (n) category

which contains those proteins that do not interact. Mathematically the categorization can be done using random probability distribution value where the continuous random variable (X) stands for the class prediction. If (X) has a higher value than the threshold (T), that is $X > T$, it is classified as positive else it is classified as negative. Outcomes for such algorithms solving binary classification problems can be either of the following four:

- True Positive (TP): Represents the case where interacting proteins are correctly classified by the algorithm to be interacting.
- True Negative (TN): In situations where non-interacting proteins are accurately classified as noninteracting pair by the algorithm.
- FP: If noninteracting proteins are wrongly pointed out to be interacting.
- FN: Represents the case where interacting proteins are erroneously classified as non-interacting pairs.

Evaluation criteria that we have used for checking the performance of the model are classification accuracy, specificity, sensitivity, precision, and F measure. These measures are defined below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{5}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{7}$$

$$\text{F} - \text{Measure} = \frac{2 * \textit{Precision} * \text{Recall}}{\textit{Precision} + \text{Recall}}. \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{9}$$

3.4 | Discussion of results

We have performed our experiment on the human PPIs dataset. We have randomly selected a total of 9484 protein pairs containing 4749 positive samples and 4735 negative samples from the original dataset. After that, we have extracted features from this sampled dataset by our method. Finally, we trained a model that considers these obtained features as input. Different measures evaluate the performances of the proposed model. Table 6 summarizes the results of threefold cross-validation with mean and SD values. The proposed approach achieves an average accuracy value of 0.8355 with an average precision of 0.8364 at a 0.8356 average F-score.

3.5 | Comparison with existing methods

Tables 7 and 8 show that the results obtained by the proposed methods are comparable with those obtained by previous best models (SAE_AC & SAE_CT[49]). The values obtained by various performance measures are mentioned in Tables 7 and 8, which are the average values of threefold cross-validation. Table 7 shows the average values of different performance measures of the models trained on the entire dataset (9484 samples). The average accuracy and average F-score values

TABLE 6 The threefold cross-validation results by proposed approach on human protein-protein interactions dataset

Test set	Precision	*F*-score	Senstivity	Specificity	Accuracy
1	0.8306	0.8335	0.8364	0.8290	0.8327
2	0.8429	0.8399	0.8370	0.8435	0.8402
3	0.8356	0.8334	0.8313	0.8359	0.8336
Mean	**0.8364**	**0.8356**	**0.8349**	**0.8361**	**0.8355**
SD	0.0050	0.0030	0.0025	0.0059	0.0033

TABLE 7 Comparison of results between the proposed approach and existing methods

Models	Avg_Pre	Avg_F-score	Avg_sens	Avg_Speci	Avg_Acc
Proposed	0.8364	**0.8356**	**0.8349**	0.8361	**0.8355**
SAE_AC[49]	**0.8395**	0.8273	0.8155	**0.8438**	0.8296
SAE_CT[49]	0.8287	0.8067	0.7860	0.8370	0.8114

TABLE 8 Comparison of results between the proposed approach and existing methods on different sample sizes

Sample size	Models	Number of positive samples	Number of negative samples	Avg_acc
	Proposed	503	497	**0.7509**
1000	SAE_AC			0.7079
	SAE_CT			0.7369
	Proposed	1250	750	**0.7968**
2000	SAE_AC			0.7893
	SAE_CT			0.7823
	Proposed	2003	997	**0.8170**
3000	SAE_AC			0.7810
	SAE_CT			0.7633
	Proposed	2833	1167	**0.8252**
4000	SAE_AC	3003	997	0.8244
	SAE_CT			0.8060
	SAE	2833	2167	0.8016
5000	SAE_AC	4003	997	**0.8496**
	SAE_CT			0.8380
	Proposed	3406	2594	0.8090
6000	SAE_AC	4250	1750	**0.8329**
	SAE_CT			0.8168
	Proposed	4250	2750	0.8256
7000	SAE_AC			**0.8325**
	SAE_CT			0.8111
	Proposed	4250	3750	**0.8236**
8000	SAE_AC			**0.8236**
	SAE_CT			0.8046
	Proposed	4749	4735	**0.8355**
9484	SAE_AC			0.8296
	SAE_CT			0.8114

obtained by our method are 0.8355 and 0.8356, respectively, which are better than those attained by the other two models (SAE_AC and SAE_CT[49]). {0.8296, 0.8114} are the average accuracy values and {0.8273, 0.8067} are the average F-score values of the earlier used models. Table 8 contains the results obtained by different models for different sample sizes. We observed that our model's accuracy values are better than those given by previous models in all cases except for the situation when the number of positive samples is much higher than the number of negative pair samples (when the sample sizes are 5000 and 6000). The possible deviation from the expected result might be because our model is consistent in predicting both the positive and negative samples. It can be inferred from the results mentioned in Table 7. The average values of sensitivity and specificity of our approach are {0.8349, 0.8361}. The average values of sensitivity and specificity of the other two models are {0.8155, 0.7860}, and {0.8438, 0.8370}, respectively. Our model outperforms other models in

terms of the results of average sensitivity. In terms of average specificity, the results predicted by our model are comparable to those obtained by the other two models.

3.6 | Statistical significance test

To show that the obtained results are statistically significant, we have performed the *t*-test at a 5% (0.05) significance level. To do this, we have run our algorithm five times using threefold cross-validation. This test gives the *P*-value, which is the probability of getting results just by chance. For results to be statistically significant at a 5% significance level, the *P*-value should be less than .05. We get the *P*-value of .025, which indicates that our results are statistically significant.

4 | CONCLUSION

In this paper, a deep learning algorithm, SAE-based classifier is used to predict whether two proteins are interacting or not. This model takes input, which is the integration of different features such as evolutionary features by PSI-BLAST tool, structural features by SPIDER2, and seven physicochemical properties. The purpose here is to study the impact of integrating multiple features that are used in the prediction of PPIs either alone or integrated with some other features. The integration of features obtained by different methods complements each other and generally enhances the classifier's performance to predict PPIs. To the best of our knowledge, this is the first work where this combination of features is used to predict PPIs. Results show that the model based on this combination of features works well to predict PPIs. The accuracy that our model gives is better than that achieved by earlier used models based on sequence-based input features. The proposed approach takes more time (for calculating the scoring matrix in case of evolutionary features) than sequence-based methods to get feature vectors. In the future, we can use deep learning-based algorithms to learn good representation from the evolutionary features instead of using it directly. We can also explore several other representations of proteins, such as Gene ontology annotations, to get feature vectors. Moreover some other classifiers will also be tried in place of SAE.

ACKNOWLEDGEMENTS

Dr. Sriparna Saha would like to acknowledge the support of Science and Engineering Research Board (SERB) of Department of Science and Technology India to carry out this research. SERB of Department of Science and Technology India, Grant/Award Number: ECR/2017/001915

CONFLICT OF INTEREST

All the authors declare that they do not have any conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available at http://www.csbio.sjtu.edu.cn/bioinf/LR_PPI/Data.htm, Reference 50.

ORCID

Kanchan Jha https://orcid.org/0000-0003-3837-3083
M. Tanveer https://orcid.org/0000-0002-5727-3697

REFERENCES

1. Zhang QC, Petrey D, Deng L, et al. Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature*. 2012;490(7421):556-560.
2. Krogan NJ, Cagney G, Yu H, et al. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*. 2006;440(7084):637-643.
3. Gavin AC, Bösche M, Krause R, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*. 2002;415(6868):141-147.
4. Ho Y, Gruhler A, Heilbut A, et al. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*. 2002;415(6868):180-183.
5. Mrowka R, Patzak A, Herzel H. Is there a bias in proteome research? *Genome Res*. 2001;11(12):1971-1973.

6. Sangaiah AK, Suraki MY, Sadeghilalimi M, Bozorgi SM, Hosseinabadi AAR, Wang J. A new meta-heuristic algorithm for solving the flexible dynamic job-shop problem with parallel machines. *Symmetry*. 2019;11(2):165.
7. Sangaiah AK, Bian GB, Bozorgi SM, Suraki MY, Hosseinabadi AAR, Shareh MB. A novel quality-of-service-aware web services composition using biogeography-based optimization algorithm. *Soft Comput*. 2019;24:1-13.
8. Sangaiah AK, Medhane DV, Han T, Hossain MS, Muhammad G. Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics. *IEEE Trans Ind Inform*. 2019;15(7):4189-4196.
9. Sangaiah AK, Arumugam M, Bian GB. An intelligent learning approach for improving ECG signal classification and arrhythmia analysis. *Artif Intell Med*. 2020;103:101788.
10. Sangaiah AK, Hosseinabadi AAR, Shareh MB, Bozorgi Rad SY, Zolfagharian A, Chilamkurti N. IoT resource allocation and optimization based on heuristic algorithm. *Sensors*. 2020;20(2):539.
11. Wang J, Gao Y, Wang K, Sangaiah AK, Lim SJ. An affinity propagation-based self-adaptive clustering method for wireless sensor networks. *Sensors*. 2019;19(11):2579.
12. Liu P, Wang J, Sangaiah AK, Xie Y, Yin X. Analysis and prediction of water quality using LSTM deep neural networks in IoT environment. *Sustain For*. 2019;11(7):2058.
13. Salman O, Elhajj IH, Chehab A, Kayssi A. A machine learning based framework for IoT device identification and abnormal traffic detection. *Trans Emerg Telecommun Technol*. 2019;e3743.
14. Sreedevi A, Rama RT. Reinforcement learning algorithm for 5G indoor device-to-device communications. *Trans Emerg Telecommun Technol*. 2019;30(9):e3670.
15. Bock JR, Gough DA. Predicting protein–protein interactions from primary structure. *Bioinformatics*. 2001;17(5):455-460.
16. Sprinzak E, Margalit H. Correlated sequence-signatures as markers of protein-protein interaction. *J Mol Biol*. 2001;311(4):681-692.
17. Zhou C, Yu H, Ding Y, Guo F, Gong XJ. Multi-scale encoding of amino acid sequences for predicting protein interactions using gradient boosting decision tree. *PLoS One*. 2017;12(8):e0181426.
18. Sriwastava BK, Basu S, Maulik U. Predicting protein-protein interaction sites with a novel membership based fuzzy SVM classifier. *IEEE/ACM Trans Comput Biol Bioinform*. 2015;12(6):1394-1404.
19. 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):3025-3030.
20. Xu B, Guan J. From function to interaction: a new paradigm for accurately predicting protein complexes based on protein-to-protein interaction networks. *IEEE/ACM Trans Comput Biol Bioinform*. 2014;11(4):616-627.
21. Liu P, Yang L, Shi D, Tang X. Prediction of protein-protein interactions related to protein complexes based on protein interaction networks. *Biomed Res Int*. 2015;2015:9.
22. Von Mering C, Krause R, Snel B, et al. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*. 2002;417(6887):399-403.
23. Ding Z, Kihara D. Computational methods for predicting protein-protein interactions using various protein features. *Curr Protoc Protein Sci*. 2018;93(1):e62.
24. Shen J, Zhang J, Luo X, et al. Predicting protein–protein interactions based only on sequences information. *Proc Natl Acad Sci*. 2007;104(11):4337-4341.
25. Altschul SF, Madden TL, Schäffer AA, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25(17):3389-3402.
26. Heffernan R, Paliwal K, Lyons 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.
27. Heffernan R, Dehzangi A, Lyons J, et al. Highly accurate sequence-based prediction of half-sphere exposures of amino acid residues in proteins. *Bioinformatics*. 2016;32(6):843-849.
28. Meiler J, Müller M, Zeidler A, Schmäschke F. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Mol Model Ann*. 2001;7(9):360-369.
29. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521:436–444.
30. Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process Mag*. 2012;29(6):82-97.
31. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*; 2012:1097-1105.
32. Chen Y, Wang J, Chen X, Sangaiah AK, Yang K, Cao Z. Image super-resolution algorithm based on dual-channel convolutional neural networks. *Appl Sci*. 2019;9(11):2316.
33. Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning; 2015. arXiv preprint arXiv:150600019.
34. Zheng HT, Han J, Chen J, Sangaiah AK. A novel framework for automatic Chinese question generation based on multi-feature neural network model. *Comput Sci Inf Syst*. 2018;15(3):487-499.
35. Silver D, Huang A, Maddison CJ, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016;529(7587):484.
36. Jha V, Savitha R, Shenoy PD, Venugopal K, Sangaiah AK. A novel sentiment aware dictionary for multi-domain sentiment classification. *Comput Electr Eng*. 2018;69:585-597.
37. Liang N, Zheng HT, Chen JY, Sangaiah AK, Zhao CZ. Trsdl: tag-aware recommender system based on deep learning–intelligent computing systems. *Appl Sci*. 2018;8(5):799.

38. Zheng HT, Chen JY, Yao X, Sangaiah AK, Jiang Y, Zhao CZ. Clickbait convolutional neural network. *Symmetry*. 2018;10(5):138.
39. Kuksa PP, Min MR, Dugar R, Gerstein M. High-order neural networks and kernel methods for peptide-MHC binding prediction. *Bioinformatics*. 2015;31(22):3600-3607.
40. Li Y, Shi W, Wasserman WW. Genome-wide prediction of cis-regulatory regions using supervised deep learning methods. *BMC Bioinform*. 2018;19(1):202.
41. Xu Y, Dai Z, Chen F, Gao S, Pei J, Lai L. Deep learning for drug-induced liver injury. *J Chem Inf Model*. 2015;55(10):2085-2093.
42. Zeng H, Edwards MD, Liu G, Gifford DK. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*. 2016;32(12):i121-i127.
43. Zhang S, Zhou J, Hu H, et al. A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res*. 2016;44(4):e32-e32.
44. Spencer M, Eickholt J, Cheng J. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Trans Comput Biol Bioinform*. 2014;12(1):103-112.
45. Wang S, Peng J, Ma J, Xu J. Protein secondary structure prediction using deep convolutional neural fields. *Sci Rep*. 2016;6(1):1-11.
46. Angermueller C, Pärnamaa T, Parts L, Stegle O. Deep learning for computational biology. *Mol Syst Biol*. 2016;12(7):878.
47. Baldi P. Autoencoders, unsupervised learning, and deep architectures. Paper presented at: Proceedings of the ICML Workshop on Unsupervised and Transfer Learning. JMLR Workshop and Conference Proceedings: Bellevue, Washington; 2012:37-49.
48. Shannon CE. A note on the concept of entropy. *Bell Syst Tech J*. 1948;27(3):379-423.
49. 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):277.
50. Pan XY, Zhang YN, Shen HB. Large-Scale prediction of human protein- protein interactions from amino acid sequence based on latent topic features. *J Proteome Res*. 2010;9(10):4992-5001.
51. Smialowski P, Pagel P, Wong P, et al. The Negatome database: a reference set of non-interacting protein pairs. *Nucleic Acids Res*. 2010;38(Suppl 1):D540-D544.

How to cite this article: Jha K, Saha S, Tanveer M. Prediction of protein-protein interactions using stacked auto-encoder. *Trans Emerging Tel Tech*. 2021;e4256. https://doi.org/10.1002/ett.4256