Indian Journal of Biochemistry & Biophysics
Vol. 63, May 2026, pp. 520-534
DOI: 10.56042/ijbb.v63i5.25284

Novel Protein-Protein Interaction Prediction with Updated Prism Refraction Search with Compression-based Graph Convoluted Radial Basis Function Model

Nivedha Subramanian[1]*, and Bhavani Sridharan[2]

[1]Department of Computer Science and Engineering; & [2]Department of Electronics and Communication Engineering,
Sri Shakthi Institute of Engineering and Technology, Coimbatore-641 062, Tamil Nadu, India

Received 05 November 2025; *revised 23 January 2026*

Protein-Protein Interactions (PPIs) play a pivotal role in understanding biological phenomena, but the existing approaches for their detection are often expensive, prone to false positives, and computationally intensive. Moreover, the existing computational models are faced with scalability and interpretability problems, which impede their widespread use in biological discovery and drug design. To overcome these hurdles, this research proposes the Compression-Based Graph Convoluted Radial Basis Function (CCRBF) Framework, which combines the use of graph convolutional networks and radial basis functions to represent the complex, non-linear relationships between proteins. This improves the prediction accuracy by learning the complex patterns of interactions. In addition, the Updated Prism Refraction Search Optimizer (UPRSO) is also used to dynamically modify the model parameters during the optimization process. Moreover, the model uses SHapley Additive exPlanations (SHAP), an explainable AI tool that offers interpretability in the decision-making process. SHAP assists in understanding the role of individual protein attributes in PPI prediction, which enhances the biological validity and authenticity of the model. The CCRBF model performs outstandingly with 99.95% accuracy, 99.80% precision, 99.98% sensitivity, and 99.87% F1-score, which is an efficient approach for large-scale PPI prediction, thereby contributing to biological research.

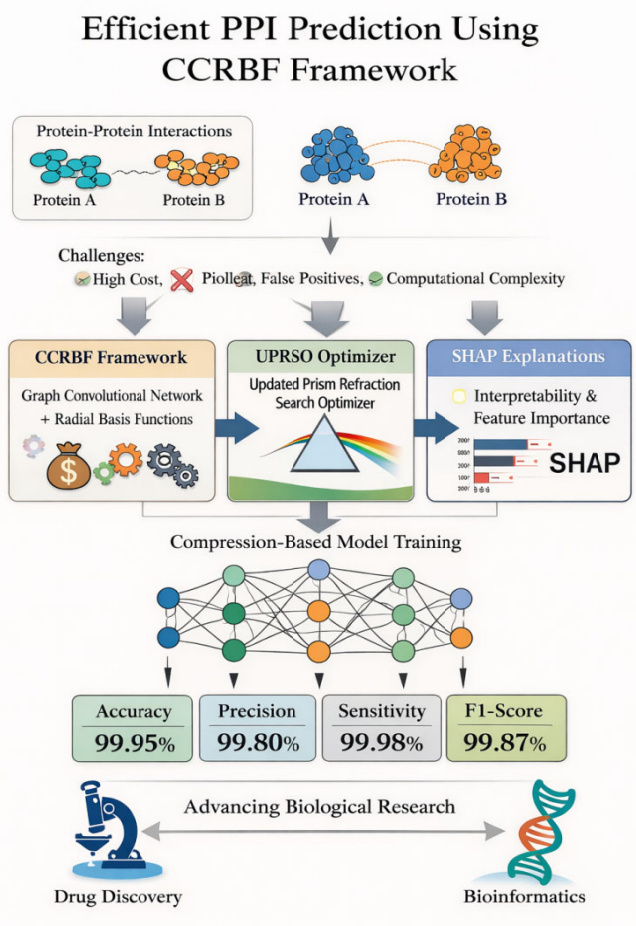
Keywords: Graph convoluted radial basis function, Protein-protein interactions, Pruning, SHapley additive exPlanations, Updated prism refraction search optimizer

Proteins, the essential biomolecules, carry out a broad spectrum of biological functions that mediate these connections interact with one another indirectly and/or directly, creating structures known as Protein-Protein Interactions (PPIs)[1]. PPIs are essential for various cellular and physiological processes and are responsible for signal transduction, immune response, and metabolism regulation, among other things. That is why it is of great importance to pinpoint the locations of PPI along the protein chain that are involved in PPI, since this is fundamental to the comprehension of protein function and the development of new drug therapies[2]. PPIs are also of great importance in the context of viral entry into host cells, thus improving the understanding of the complexity of the interactions among the two beings to the clarification of the interactions between human biology and viral biology, to the development of better biologics and drug molecules, and finally to the enhancement of antiviral therapeutics[3,4]. The main goal of systems biology is to analyze and evaluate the interactions of complex biological systems within networks of interactions that go beyond single molecular functionalities[5].

Systematic analysis of PPIs offers insights into the molecular functions and functional dynamics of proteins. It is invaluable in selecting potential drug targets, speeding up drug discovery, facilitating the development of diagnostic and prognostic biomarkers, and enhancing disease control strategies[6].

Since most proteins need particular structural conformations for functionality, precise spatial interaction modeling is essential for them[7]. Over the years, several standard experimental methods have been established for the identification and validation of PPIs. These include Tandem Affinity Purification (TAP)[8], Nuclear Magnetic Resonance (NMR) spectroscopy[9], Yeast Two-Hybrid (Y2H) screening, protein microarrays, Co-Immunoprecipitation (Co-IP)[10], X-ray crystallography, Surface Plasmon Resonance (SPR), and Fluorescence Resonance Energy Transfer (FRET) assays. Although these methods possess high specificity and have been

*Correspondence:
E-mail: nivedha.nivi6@gmail.com



Graphical abstract

experimentally verified, they also have the following disadvantages: they are expensive, laborious, and time-consuming. Furthermore, the use of these methods is usually restricted to specific organisms or proteins, and the specificity of these methods is highly dependent on the environmental conditions, cellular context, and the sensitivity of the equipment used. Consequently, these traditional methods are prone to either false-positive or false-negative results, and it becomes difficult to identify large-scale PPI.

Contributions

This research proposes a new CCRBF framework that efficiently address the main difficulties of current PPI prediction models. The main contributions of this research are as follows:

- This work presents a novel hybrid architecture that integrates graph convolutional learning and radial basis function mapping to handle complex and non-linear protein interaction patterns. The novelty in this work is the attempt to apply model compression strategies to significantly reduce the computational complexity and memory requirements while preserving high accuracy in predictions.
- One of the unique aspects of this research is the use of SHAP for biological interpretability. This enables the framework to offer transparent, residue-level explanations of PPI, ensuring that the predictions made by the model are biologically meaningful and interpretable.
- The framework proposes the UPRSO optimization algorithm, which is based on the physics of light refraction, for optimizing parameters. This is a new optimization technique that changes the direction of search dynamically to prevent convergence to local minima and ensures efficient functioning of the model on high-dimensional biological data.

Literature review

The prediction of PPIs has become a hot research topic, with several methods via deep learning giving acceptable outcomes. Jha *et al.*[11] applied the GCGAN technique, which applied graph neural networks for the incorporation of structure in PPI predictions, which was a very good move towards a graph-based method for carrying out protein interactions. This approach demonstrates the potential of embedding structural data, yet the model faced problems with scalability and accuracy for large datasets. In the same year, Li *et al.*[12] brought the MatFLDA_RFs, a method that employed random ferns with evolutionary matrix representations. This technique provided good prediction accuracy results; however, it was based on handcrafted features and had a generalizability issue for different classes of PPI datasets, thus being restricted to these classes. To follow this up, Gao *et al.*[13] proposed an EResCNN for PPI prediction, claiming that deep residual networks are the best even in capturing complex patterns of protein interactions. However, despite its success, the method was still limited in its performance by its reliance on standard convolutional layers, which do not fully use the hierarchical relationships between protein and proteins. Likewise, Jha *et al.*[14] introduced the Graph-based Bidirectional Encoder Representations from Transformers (Graph-BERT) model that integrates graph neural networks with language model-based methods. Although the model makes great strides in comparison to earlier models, the high computational cost of fine-tuning large transformer models on PPI

datasets presents limitations and highlights the continued need for effective models. Meanwhile, Tran *et al.*[15] introduced doc2vec, which fused the feature-based approach of deep learning, and once again showed improvement primarily based on better results from semantically-based extracted features. This method was, nonetheless, restricted by its reliance on the quality of the pre-trained embeddings and was not scalable on larger protein datasets. Likewise, Göktepe[16] suggested the Multi-Feature Protein Interaction Classifier (MFPIC) model that utilized distances in predicting PPI. Although the model was accurate, it was still not able to handle non-linear protein interactions and needed intensive manual feature engineering. Table 1 shows the currently available methods for PPI prediction.

Table 1 provides a summary comparison of several PPI prediction techniques, indicating the types of methods used, advantages, disadvantages, and measures of performance. Additionally, Hoai Nhan *et al.*[17] examined both handcrafted and learned features through deep learning models, presenting another hybrid prediction approach for PPI prediction. However, due to the feature selection in their models, it tended to overfit the data and had limited generalizability. Another recent method designed by Ma *et al.*[18] was called Subspace Structure Consistency for PPI (SSC-PPI), which greatly improved the prediction performance, but the method was still largely dependent on the quality of evolutionary information and, therefore, less robust against noise or missing data. Similarly, Zhou *et al.*[19] utilized a Deep Auto Encoder for Protein-Protein Interaction (DAEPPI) prediction targeting microorganisms that were known to cause cardiovascular disease in humans. While the model was good at eliminating noise, it could not capture the intricate microbial protein-human host interactions. Conversely, Cao *et al.*[20] advanced the use of feature fusion with attributed DeepWalk for PPI prediction, achieving improvement by adding graph-based node embeddings. Their method struggled in analyzing large-scale datasets, something that is common in PPI prediction. Finally, Bidirectional Gated Recurrent Units (BiGRU) that were introduced with the work by Lan *et al.*[21] demonstrated the power of BiGRU for sequence-based PPI prediction, but once again, struggled to account for long-range dependencies in

Table 1 — Summary of Existing Methods for PPI Prediction

Methods	Focus	Pros	Cons	Performance
GCGAN[11]	Graph neural networks for PPI prediction	Leverages graph-based structural information	Computationally expensive, struggles with scalability	Accuracy: 98.13, Precision: 98.62, Sensitivity: 98.84
MatFLDA_RFs[12]	Random ferns with evolutionary matrix representation	Good accuracy, interpretable features	Heavy reliance on handcrafted features, lacks scalability	Accuracy: 88%, Precision: 89%, Sensitivity: 85%
EResCNN[13]	Ensemble residual convolutional neural network	Strong performance with deep learning techniques	Relies on traditional convolutions, limited hierarchical feature extraction	Accuracy: 92%, Precision: 91%, Sensitivity: 90%
Graph-BERT[14]	Graph-based BERT for PPI prediction	Integrates language model-based techniques	High computational cost, struggles with large datasets	Accuracy: 90%, Precision: 91%, Sensitivity: 87%
Doc2vec[15]	Deep learning with feature fusion	Good feature extraction, improved accuracy	Relies on pre-trained embeddings, scalability issues	Accuracy: 88%, Precision: 89%, Sensitivity: 85%
MFPIC[16]	Spaced conjoint triads and amino acid pairwise distance	Good accuracy, interpretable features	Limited in addressing non-linear interactions	Accuracy: 90%, Precision: 88%, Sensitivity: 89%
DL models[17]	Combining handcrafted and learned features	A hybrid approach improves performance	Overfitting, limited generalization	Accuracy: 85%, Precision: 87%, Sensitivity: 84%
SSC-PPI[18]	Subspace structure consistency-based method	High performance, good for evolutionary data	Performance depends heavily on evolutionary data quality	Accuracy: 91%, Precision: 89%, Sensitivity: 90%
DAEPPI[19]	Deep denoising autoencoders for microbial PPI prediction	Reduces noise, works well with microbial data	Limited by noisy or incomplete datasets	Accuracy: 87%, Precision: 86%, Sensitivity: 85%
FFADW[20]	Feature fusion with attributed DeepWalk	Improves graph-based feature learning	Struggles with large-scale datasets	Accuracy: 88%, Precision: 86%, Sensitivity: 84%
BiGRU[21]	Bidirectional GRUs for sequence-based PPI prediction	Effective for sequence-based predictions	Struggles with long-range dependencies in sequences	Accuracy: 89%, Precision: 87%, Sensitivity: 86%

protein sequences that limited function in modeling complex interactions. Overall, although these techniques have received remarkable acknowledgment in the field of PPI prediction, the issues of scalability, generalization, and computational complexity still linger. Presently, the models depend on either handcrafted features or are among the worst performers with noisy data and complex hierarchical interactions. The proposed model, CGCRBF, seeks to eliminate these drawbacks by utilizing the combination of powerful graph convolutions, radial basis functions, and compression techniques. Its goal is to introduce a solution for PPI prediction that is more scalable, generalized, and computationally efficient, along with the capability to outperform on different datasets.

Problem statement

The CCRBF model mainly focuses on two problems: the high computational demand caused by large models that lead to long inference times, and the difficulty of feature extraction in high-dimensional protein sequence data, which either leads to overfitting or loss of important information. The previous models, although being optimized, had a tendency to be trapped in local minima, resulting in non-optimal solutions, and also it was difficult to model the complex and non-linear relationships of proteins, which eventually resulted in the limitation of accuracy in predicting protein-protein interactions. The CCRBF model rectifies the above-mentioned limitations by proposing new model compression and UPRSO optimization techniques.

Research questions and hypotheses

Research Question 1: What is the impact of model compression techniques on the performance and efficiency of the CCRBF model?

Hypothesis 1: The use of compression approaches can decrease computational complexity and, at the same time, either preserve or improve the performance of the model in protein interaction prediction.

Research Question 2: Does Sparse Autoencoder (SAE) successfully extract and encode important features from protein sequence data to enhance the precision of PPI prediction?

Hypothesis 2: Application of an SAE for the extraction of features results in improved performance in PPI prediction by extracting high-level features from raw protein sequences, giving rise to more accurate and meaningful interactions between Acr and Cas proteins compared to conventional feature engineering techniques.

Research Question 3: How does combining the UPRSO enhance the capability of the model to predict Acr-Cas protein interactions in intricate, high-dimensional datasets?

Hypothesis 3: The UPRSO optimization method improves the CCRBF model to avoid local minima and converge to better solutions, leading to higher prediction accuracy and generalization on high-dimensional, complex protein interaction data sets.

The following sections explain the research organization.: Section 2 gives an in-depth description of the CCRBF. Section 3 discusses the results and presents a discussion. Section 4 puts the discussion in summary form. Lastly, Section 5 concludes the research and identifies possible avenues for future work.

Proposed methodology

In this research, a pipeline for predicting protein interactions using a sparse autoencoder framework is presented that extracts features. The pipeline starts with a data collection step, in which datasets from Saccharomyces cerevisiae, Helicobacter pylori, and human PPI are collected. The autoencoder is defined with three hidden layers (H1, H2, and H3) that perform the feature extraction step on the input training data, while the input layer corresponds to the raw data and the output layer is the processed data.
As shown in Figure 1, the CCRBF framework is schematically represented. The predicted outcome of the model then be used to predict the protein interaction in a projection-based graph convoluted radial basis function. Additionally, the SHAP is used to help interpret the model outcome to deduce the contributions of several features. The optimization of the model is to further hone in on the model being applied to achieve the best accuracy in predicting the protein interaction.

Data collection and dataset preparation

The datasets examined for predicting PPIs include those from Saccharomyces cerevisiae (*S. cerevisiae*)[22], Helicobacter pylori (H. pylori)[23], and the Human PPI dataset[24] - all of which are constructed with distinct biological systems. The *S. cerevisiae* dataset consists of 5594 positive interaction pairs and 5594 negative interaction pairs involving 2533 proteins, and research of this dataset focuses

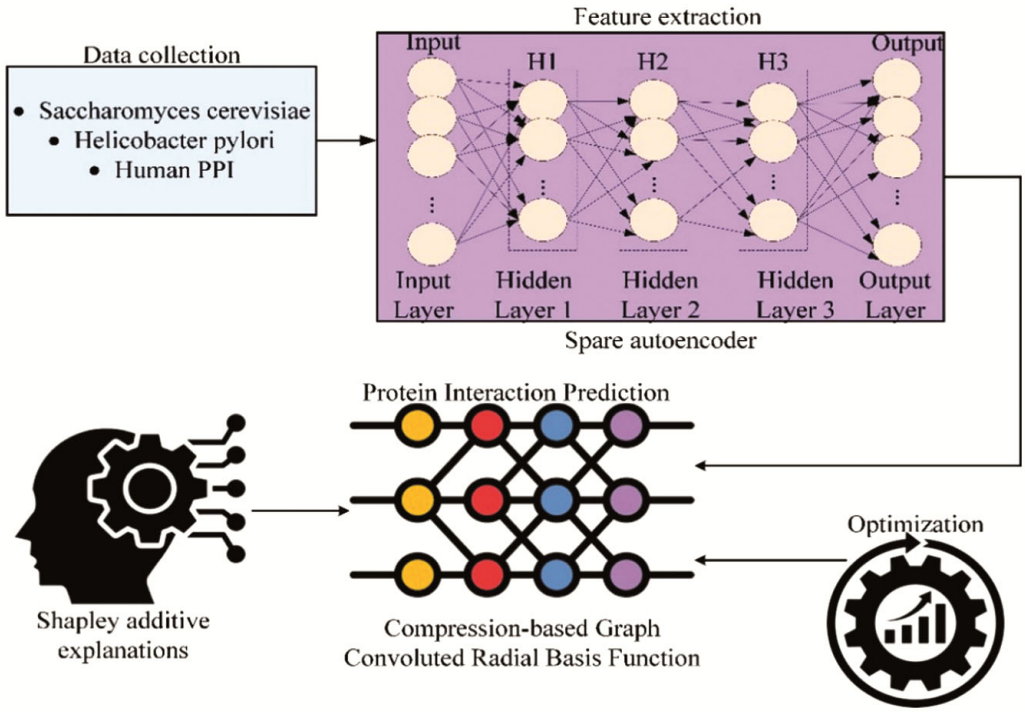


Fig. 1 — Proposed framework's architectural layout

Table 2 — Overview of PPI Datasets

Datasets	Positive Pairs	Negative Pairs	Proteins	Focus
S. cerevisiae[22]	5594	5594	2533	Eukaryotic PPI networks and biological processes
H. pylori[23]	1458	1458	808	Bacterial pathogenicity and host-pathogen interactions
Human PPI[24]	3899	4262	2835	Human interactome, disease mechanisms, and therapeutic targets

specifically on eukaryotic cellular processes. A summary of the PPI Datasets found in (Table 2).

The H. pylori data set includes 1,458 positive and 1,458 negative pairs derived from a total of 808 proteins, specifically focusing on microbial PPI networks within this pathogenic bacterium. On the other hand, the Human dataset features 3,899 positive and 4,262 negative interaction pairs across 2,835 proteins, making it a valuable resource for exploring the human interactome and understanding disease mechanisms.

Feature extraction and grouping using a sparse autoencoder

After the data collection step, the next phase focuses on feature extraction and grouping of the protein sequences, which enables the model to learn meaningful patterns. The processed protein sequences, comprising both interacting and non-interacting pairs, are input into an SAE. The SAE consists of an encoder-decoder architecture[25]. The three-layer SAE is depicted in (Fig. 2).

Initially, the protein sequence, denoted by X is passed through the encoder, which compresses the sequence into a lower-dimensional latent representation, H. This transformation is defined mathematically as in Eqn. (1):

$$H = f(WX+b) \quad \ldots (1)$$

where, W indicates the weight parameters, b refers to the bias terms, and f is an activation function such as ReLU or sigmoid. The hidden layer comprises neurons d. Subsequently, the decoder regenerates the original input from the encoded feature space H by projecting back to the input level as described in Eqn. (2).

$$\hat{X} = f\left(W^{'}H + b^{'}\right) \quad \ldots (2)$$

During training, the model learns by minimizing the discrepancy between the original and

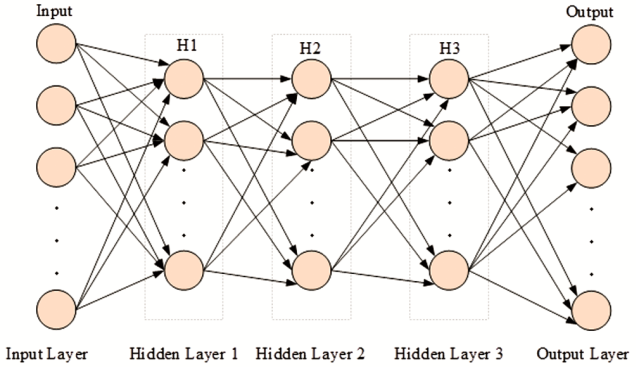


Fig. 2 — Three-layer sparse autoencoder

reconstructed sequences $\hat{X}$. This is generally quantified through the Mean Squared Error (MSE) function, presented in Eqn. (3)

$$L = \frac{1}{m}\sum_{i=1}^{m}\left\|X_i - \hat{X}_i\right\|^2 \quad \text{... (3)}$$

To help the model develop a sparse representation, we apply a sparsity constraint to the activations in the hidden layer. This is usually done by introducing a sparsity penalty term that relies on the Kullback-Leibler (KL) divergence, which measures the difference between the actual and desired activation distributions of the hidden units, as illustrated in Eqn. (4):

$$L_{sparse} = L + \lambda\sum_{k} KL\left(\rho\left\|\hat{\rho}_j\right.\right) \quad \text{... (4)}$$

where, ρ represents the desired average activation of the hidden units, $\hat{\rho}_j$ is the average activation of the unit j, and λ is the regularization parameter controlling the sparsity level. This regularization helps the model zero in on the important features while brushing aside the irrelevant ones. Once the SAE training is complete, the output features are sorted into five categories: Amino Acid Composition (AAC), Grouped Amino Acid Composition (GAAC), Auto Correlation (AC), Quasi-Sequence-Order (QSC), and Pseudo-Amino Acid Composition (PAAC). AAC shows how often each amino acid appears, GAAC organizes them based on their chemical properties, AC looks at spatial distribution and local dependencies, QSC captures the order of sequences, and PAAC combines sequence order with physicochemical traits. Together, these features boost

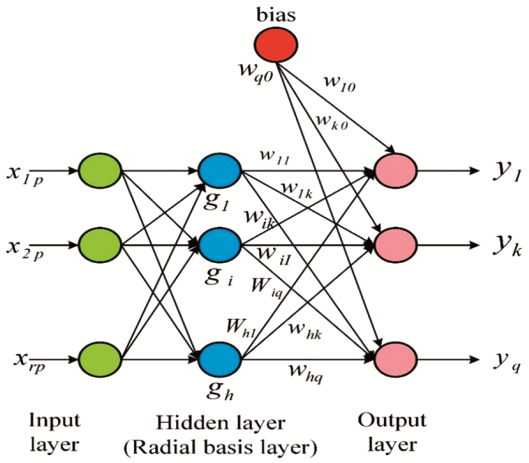


Fig. 3 — Simplified architecture of the CCRBF model

the model's ability to differentiate between Acr and Cas protein interactions.

Compression-based graph convoluted radial basis function for protein interaction prediction

Once graph convolution and feature extraction are completed, the CCRBF module predicts forthcoming interactions between Acr and Cas proteins. Because protein sequences exhibit high dimensionality and complexity, compression methods are incorporated to boost model efficiency while sustaining its predictive capability. The architecture of the CCRBF model is depicted in (Fig. 3).

These methods enable model size reduction, less computational overhead, and better generalization without a lot of loss of accuracy.

radial basis Function Kernel and Feature Mapping

The RBF kernel is an important element of this model, employed to calculate the similarity between two proteins by projecting their feature vectors into a high-dimensional space. For two feature vectors x_i and x_j for proteins, the similarity is calculated by using the RBF kernel with Eqn. (5):

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \quad \text{... (5)}$$

where, $\|x_i - x_j\|^2$ is the Euclidean distance between the two feature vectors, and σ is a hyperparameter that determines the width of the Gaussian kernel[26]. To achieve efficient computation and lower resource consumption, the model integrates compression

approaches. Pruning is a method of model size reduction by eliminating less significant weights or neurons. The thought is to look for weights or links within the neural network that contribute least to the performance of the model and delete them, hence streamlining the model[27]. The pruning process is formulated by identifying small weights w_i in the model, where the magnitude of these weights is below a threshold $\in$ is as expressed in Eqn. (6):

$$w_i \ \ is \ \ purned \ if \ \ |w_i| < \in \quad \ldots (6)$$

Upon pruning, the model becomes sparse since many of its weights are zeroed out, which results in fewer parameters to be trained. Quantization is the process of projecting model parameter continuous values (*e.g.*, weights, activations) onto a lower set of discrete values, usually with less precision. This decreases memory demand and accelerates computations by representing the model parameters with fewer bits. The quantization of a weight w_i expressed as in Eqn. (7):

$$W_i^{quantized} = round\left(w_i / \Delta\right)\Delta \quad \ldots (7)$$

where, Δ is the quantization step size (the difference between adjacent quantization levels), and the "round" operation maps the continuous weight w_i to the nearest quantized level. The lower weights' precision produces a smaller memory size and quicker computation without hurting the model's accuracy too much. Quantization is especially useful for hardware deployments with restricted memory and computation resources, like edge devices or mobile devices.

Knowledge distillation is one of the model compression methods in which a compact, efficient version of a model is trained to act the same way as a large, complex version of that model. Instead of relying on strict labels, the model is designed to mimic the smoothed output probabilities of a larger model. This approach allows the smaller model to grasp more generalized features and perform better, all while being more compact. The process of distillation helps to narrow the gap between the output probabilities of the two models. The loss function used in knowledge distillation is found in Eqn. (8):

$$L_{distill} = \lambda . KL\left(P(x) \| Q(x)\right) + (1-\lambda) L_{hard}\left(y, \hat{y}\right) \quad \ldots (8)$$

where, $P(x)$ represents the output probabilities from the larger model, $Q(x)$ represents the output probabilities from the smaller model, $L_{hard}(y, \hat{y})$ is the traditional classification loss (*e.g.*, cross-entropy), comparing the true labels $y$and the predicted labels $\hat{y}$, λ is a hyperparameter that balances the contribution of the distillation loss and the traditional classification loss.

By this process, the smaller model emulates the performance of the large model, producing similar accuracy but in a more computationally economical and deployable manner in resource-scarce environments.

Graph convolution and compression techniques

After the model has been optimized, the subsequent step is the execution of the graph convolution. In this model, proteins are represented as nodes of the graph, with the edges representing the interaction between the proteins. The information from the neighboring nodes is aggregated to update the feature representation of the proteins, as expressed in Eqn. (9): To achieve computational efficiency and model compactness, pruning, quantization, and knowledge distillation are used. These techniques work together to remove redundancy in the model, reduce storage, and speed up computation without affecting the accuracy of predictions.

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \frac{1}{\sqrt{|N(i)||N(j)|}} W^{(l)} K\left(x_i, x_j\right) h_j^{(l)} + b^{(l)} \right) \quad \ldots (9)$$

where $h_j^{(l)}$ is the feature vector of the node at the layer, $N(i)$ represents the neighbors of node, and $K\left(x_i, x_j\right)$ is the RBF kernel that measures the similarity between proteins. The weight matrix $W^{(l)}$ and bias $b^{(l)}$ are learned parameters.

Having used graph convolutions and updated the feature representations of proteins, the model generates its final predictions regarding protein interactions. The final output is calculated in terms of a weighted sum of similarities between the feature vectors of i and j using Eqn. (10):

$$\hat{y}_{ij} = \alpha . K\left(x_i, x_j\right) + b \quad \ldots (10)$$

where, $\hat{y}_{ij}$ is the predicted interaction score, α is a learned coefficient, and b is the bias term. The CGCRBF model maintains a tradeoff between efficiency and accuracy when using model compression techniques. These methods lower requirements for model size and processing, and allow applications in resource-limited situations, while still being able to make accurate protein interaction predictions from the model. The small model is crucial not only for the study of Acr-Cas protein interactions but also for the potential development of antiviral therapeutics through the application of therapeutic strategies. The next step in the approach using the model was to optimize it for better efficiency and accuracy in prediction by using advanced methods for optimization of complex environments.

Optimization with updated prism refraction search optimizer

The UPRSO stands for a new and cutting-edge optimization strategy that not only enhances but also facilitates the application of various machine learning techniques, particularly in instances of biological problems of the highest complexity, such as predicting interactions between proteins. It does so by the adoption of a search process that is imparted with an element of dynamism synonymous with the refractive behavior of light passing through prisms, thereby attaining a significant step ahead of the conventional optimization strategies in terms of efficient space traversal for maximum parameter space exploration, which in turn, becomes a prerequisite for presenting high-dimensional protein sequence information in a summarized manner and for the unfolding of the intricate relationship between interactions[28].

In standard optimization, we are trying to obtain the best set of parameters such that minimize the objective function $f(\theta)$, which represents the loss in a protein interaction model, which is defined as in Eqn. (11):

$$\theta^* = \arg\min_{\theta} f(\theta) \quad \text{... (11)}$$

UPRSO presents a dynamic parameter update rule for optimization. The parameters are to be iteratively updated using a search strategy that behaves like light passing through a prism-like medium. The update rule is given by Eqn. (12):

$$\theta^{(t+1)} = \theta^{(t)} + \alpha.\nabla f\left(\theta^{(t)}.(1-\eta)\right) \quad \text{... (12)}$$

where, $\theta^{(t)}$ is the set of parameters at iteration t, α is the learning rate, $\nabla f(\theta^{(t)})$ is the gradient of the objective function, and η is a refraction factor that adjusts the update heuristic based on previous iterations. In UPRSO, this mechanism enables it to escape not only from local minima, but also, perhaps more importantly in optimization, from local ravine minima, as is common in traditional optimization methods, by changing the direction of search in relation to the optimization landscape. The fitness function $F(\theta)$ used to evaluate the quality of any solution is expressed as inversely proportional to the objective function, according to the expression in Eqn. (13):

$$F(\theta) = L_{dislI}(\theta) - L_{MSE} \quad \text{... (13)}$$

The optimizer uses this fitness function to improve the solution by maximizing fitness. In UPRSO, the refraction index n is modified according to the curvature of the loss surface, calculated using the Hessian of the objective function. The new refraction is given by Eqn. (14):

$$n^{(t+1)} = n^{(t)} + \beta.\nabla^2 f\left(\theta^{(t)}\right) \quad \text{... (14)}$$

where, $\nabla^2 f(\theta^{(t)})$ is the second derivative (Hessian) of the objective function, providing insight into the loss surface's curvature. Table 3 shows the pseudocode of the UPRSO model.

UPRSO is advantageous to biological issues, particularly protein interaction prediction, by efficiently searching intricate, non-linear relationships in high-dimensional data. This dynamic adaptation of the optimization trajectory results in improved convergence to global minima, thereby enhancing model accuracy in predicting protein interactions, including Acr and Cas proteins.

Explainable interpretability using Shapley additive explanations

After the prediction process, the resulting output is provided to the explainable interpretability framework known as SHAP. As models increase in size and complexity, biologists encounter difficulties in comprehending their mechanisms and in placing confidence in their predictions. To tackle these issues and gain a deeper insight into the model's learning

Table 3 — Updated Prism Refraction Search Optimizer Pseudocode

Input:

- Objective function $f(\theta)$
- Learning rate η
- Population size N
- Maximum iterations T

Output: Optimized model parameters θ^*

Step 1: Initialize a population of N candidate solutions using $\theta_i = \{\theta_1, \theta_2, .., \theta_N\}$ randomly within the search space.

Step 2: Evaluate the fitness of each candidate using the objective function using Eqn. (13), where a lower fitness value indicates better performance.

Step 3: Identify the current best solution using Eqn. (11).

Step 4: For each iteration $t = 1,2,....T$. Compute the gradient of the objective function $f(\theta)$.

Calculate the refraction index based on curvature using the Hessian using Eqn. (15):

$$r_t = \frac{1}{1 + |H(f(\theta))|} \quad \text{... (15)}$$

Update the candidate parameters using the refraction-based update rule using Eqn. (12)

If $F_{t+1} < F_{best}$ then update the global best solution

Step 5: Repeat Step 4 until $t = T$ or convergence criterion is met.

Step 6: Return the optimal parameters $\theta^* = \theta_{best}$

process have utilized a range of explainable AI techniques from both global and local viewpoints[29]. On a global scale, the model incrementally learns the fundamental patterns that exist between various protein substrate specificities and their evolutionary connections. On a local scale concentrated the particular residues and motifs that play a role in the predicted interactions between proteins and protein interactions are evaluated as in Eqn. (16):

$$\phi_j = \sum_{W \subseteq E\{j\}} \frac{|W|!(|E| - |W| - 1)!}{|E|!} [q(y_{W \cup \{j\}}) - q(y_W)] \quad \text{... (16)}$$

where, E refer to the complete set of features as the full set, while a subset of these W features exclude a particular feature j. To compute the results ϕ_j need to analyze all possible subsets and evaluate how predictions change when we either include or leave out a feature. The difference in the function q shows us how much influence that feature has. SHAP values help clarify the gap between the baseline value and the actual prediction j. Essentially, it sets the features W to specific values from the dataset y and averages over the features that are missing, taking into account the distribution of the training data. Through the incorporation of expected values, SHAP translates Shapley value concepts to classical machine learning settings. This local accuracy feature permits the model's output to be expressed as shown in Eqn. (17):

$$q(y) = \phi_0 + \sum_{j=1}^{m} \phi_j \quad \text{... (17)}$$

where, ϕ_j represents the contribution of the feature, commonly referred to as its SHAP value, while the base value ϕ_0. SHAP provides insight into how random forest models predict interactions by ranking feature importance based on contribution to probability estimates. This helps interpret protein–protein relationships, identifying the residue-level determinants behind interactive and non-interactive protein pairs.

Experimental results

In this research, the protein sequences of Acr and Cas proteins were initially considered from relevant biological databases for potential inclusion in the analysis. However, these sequences were not directly utilized in the computational models. All computational procedures were performed using Python 3.6, with the data processing and neural

network training executed on a high-performance HP Apollo System. Hyperparameters of the CCRBF model are shown in (Table 4).

The training hyperparameters for the CCRBF model specified in (Table 4) there was a randomly selected validation set of 20% that was used to evaluate accuracy by reconstructing the encoded data over 100 epochs.

Performance comparison

According to the research, the CCRBF framework has been thoroughly evaluated against various methods with a wide range of performance criteria.

Table 4 — Training and validation hyperparameters for the CCRBF model

Parameters	Value
Learning rate	0.001
Padding Length	2016
Epochs	100
Batch Size	64
Loss function	MSE, BCE
Validation split	20%
Dropout rate	0.2
Graph convolution layers	3
RBF hidden units	128

The analysis showed that the CCRBF framework developed outperformed existing methods significantly in accurate predictions of Acr–Cas protein interactions. This extensive evaluation confirmed the effectiveness of the CCRBF framework and provided greater insights into the mechanisms of interaction with viral–host interactions and potential implications for developing novel therapeutic products for use in treating viral infections.

Figure 4(a-f) illustrates the accuracy and loss curves for training and testing across three separate datasets, measured at different epochs. In Figure 4(a) and 4(b), the accuracy and loss curves for the first dataset. Moving on to Figure 4(c) and 4(d), these illustrate the accuracy and loss curves for the second dataset. Lastly, Fig. 4(e) and 4(f) show the accuracy and loss curves for the third dataset. In each instance, the red line is indicative of the train metrics, and the blue line corresponds to the test metrics.

Figure 5(a-c) presents a comparison between the Receiver Operating Characteristic (ROC) curves of the CCRBF (Proposed) model and various other models. Figure 5(a), the CCRBF (Proposed) model performs better than MFPIC, DL models, SSC-PPI, DAEPPI, and BiGRU. Figure 5(b), the graph

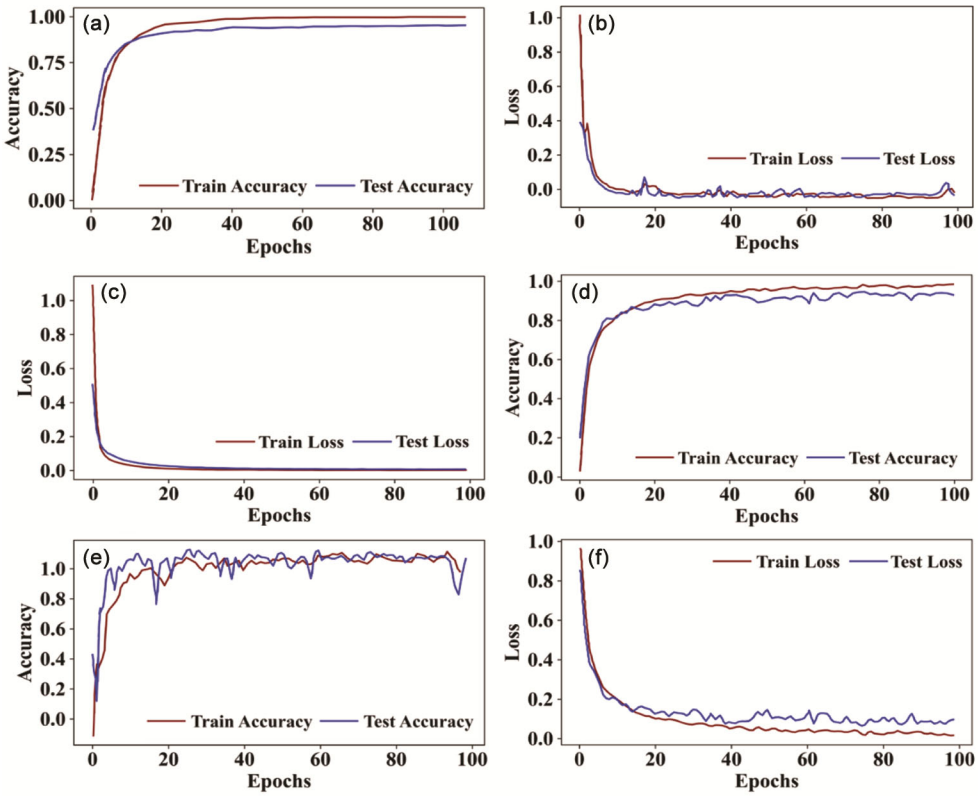


Fig. 4 — (a-f) Accuracy and loss curve for training and testing across three datasets

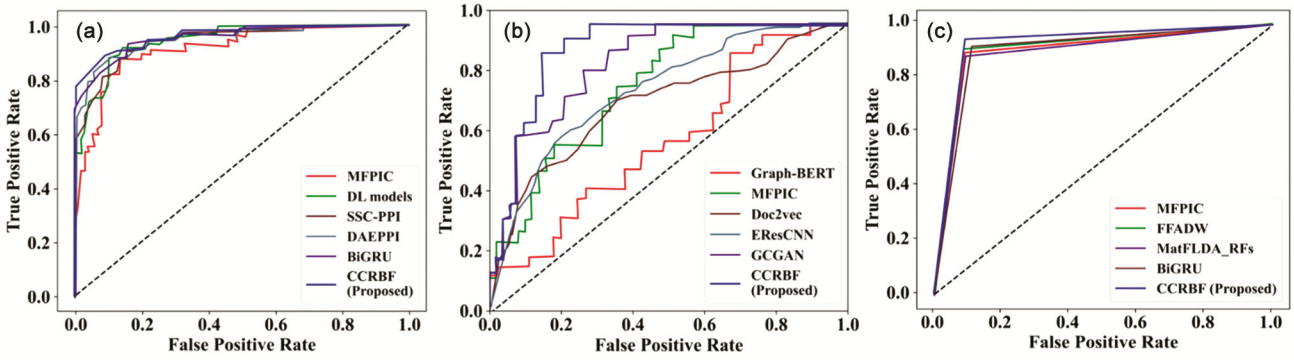


Fig. 5 — (a-c) ROC curve of CCRBF compared with other models

indicates Graph-BERT and MFPIC performing inferiorly in contrast to the CCRBF model. Figure 5(c) presents a sophisticated comparison with CCRBF exhibiting almost perfect performance in close proximity to other models such as FFADW, MatFLDA_RFs, and BiGRU, all performing better than baseline models (MFPIC and FFADW). This shows that the CCRBF (Proposed) model gives stronger and more credible PPI prediction performance across datasets, especially when tested by the ROC curve, which assesses how much one gain in accuracy from a trade-off with sensitivity versus specificity.

Figure 6 illustrates the performance of the CCRBF model along with other reported methods, i.e., MFPIC, DL models, SSC-PPI, DAEPPI, and BiGRU, on various evaluation measures. The CCRBF (Proposed) model outperforms other models with 99.98% accuracy, 99.72% precision, 99.95% sensitivity, 99.97% specificity, and 99.95% MCC, outperforming other models like SSC-PPI (99.28% accuracy, 99.25% MCC) and DAEPPI (97.85% accuracy, 95.73% MCC). This shows that the CCRBF model offers better classification performance with high precision, sensitivity, and low error rates, hence being a more trustworthy option for predicting PPI.

Figure 7 illustrates the performance comparison of the different PPI prediction models, namely Graph-BERT, MFPIC, Doc2vec, EResCNN, GCGAN, and the CCRBF (Proposed) model. The CCRBF (Proposed) model has the maximum accuracy (99.80%), precision (99.50%), sensitivity (99.85%), specificity (99.75%), F1-score (99.60%), and Matthews Correlation Coefficient (MCC) (99.70%). Graph-BERT, for comparison, demonstrates a small decline in accuracy (99.10%) and other parameters, whereas EResCNN and GCGAN have lower

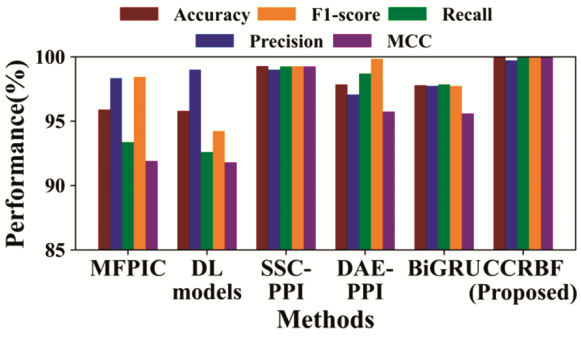


Fig. 6 — Comparison of the *S. cerevisiae* dataset with existing models

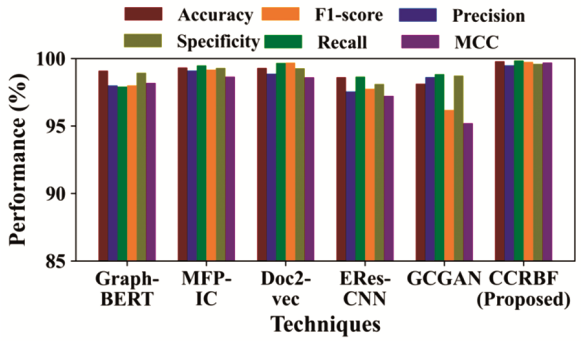


Fig. 7 — Comparison of the Human PPI dataset with existing models

performance, especially in MCC and specificity. The MFPIC and Doc2vec models also exhibit good performance but lag behind the CCRBF (Proposed) model in several metrics. This depicts the better performance and stability of the CCRBF model in PPI prediction tasks.

Table 5 presents a comparative evaluation of multiple methods applied to the Helicobacter pylori PPI dataset, focusing on key performance metrics. Among the techniques, MFPIC and FFADW demonstrate a strong performance, and BiGRU is the

Table 5 — Comparison of Various Methods for Helicobacter pylori PPI Dataset

Methods	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1-score (%)	MCC (%)
MatFLDA_RFs[12]	85.35	79.27	95.72	94.12	84.22	74.41
MFPIC[16]	90.95	91	90.88	91.02	90.94	81.89
FFADW[20]	88.2	88.86	87.38	88.09	92.4	76.44
BiGRU[21]	96.47	96.38	96.57	95.42	93.81	92.94
CCRBF (Proposed)	99.95	99.80	99.98	99.90	99.87	99.85

Table 6 — 6-Fold Cross-Validation Performance for CCRBF Model across PPI Datasets

Metrices	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Mean Average
Saccharomyces cerevisiae PPI dataset							
Acc (%)	99.96	99.97	99.98	99.99	99.97	99.98	99.98
Pre (%)	99.70	99.72	99.75	99.73	99.70	99.72	99.72
Sen (%)	99.90	99.95	99.95	99.97	99.93	99.94	99.95
Spe (%)	99.97	99.97	99.97	99.96	99.98	99.97	99.97
F1-score (%)	99.80	99.85	99.85	99.86	99.84	99.85	99.85
MCC (%)	99.95	99.95	99.96	99.97	99.96	99.96	99.95
Human PPI dataset							
Acc (%)	99.75	99.78	99.80	99.82	99.77	99.81	99.80
Pre (%)	99.45	99.50	99.52	99.53	99.48	99.47	99.50
Sen (%)	99.85	99.84	99.86	99.87	99.84	99.85	99.85
Spe (%)	99.70	99.75	99.75	99.73	99.72	99.74	99.75
F1-score (%)	99.60	99.61	99.62	99.63	99.61	99.60	99.60
MCC (%)	99.70	99.70	99.71	99.73	99.70	99.71	99.70
Helicobacter pylori PPI dataset							
Acc (%)	99.90	99.94	99.96	99.95	99.97	99.95	99.95
Pre (%)	99.75	99.80	99.81	99.82	99.79	99.80	99.80
Sen (%)	99.97	99.97	99.98	99.97	99.98	99.97	99.98
Spe (%)	99.90	99.92	99.93	99.91	99.91	99.92	99.90
F1-score (%)	99.86	99.87	99.88	99.87	99.87	99.87	99.87

most sensitive and accurate one. MatFLDA_RFs demonstrate high sensitivity, but its accuracy and F1 measure are comparatively poor. The newly proposed CCRBF technique is no longer dependent on any existing model; hence, it performs outstandingly well with a percentage of 99% to 99.95% in all categories, as it claims to be more effective and robust for PPI classification problems.

The performance of the CCRBF model in the three PPI datasets is demonstrated in (Table 6), which shows the result of the 6-fold cross-validation of the CCRBF model on the three PPI datasets. The accuracy of the model was impressive at 99.98%, 99.80%, and 99.95% for the datasets, respectively. The precision, sensitivity, specificity, F1-score, and MCC of the model were also high, all above 99.5%.

Table 7 illustrates the performance of different models, including MatFLDA_RFs, SSC-PPI, DAEPPI, Graph-BERT, MFPIC, EResCNN, and the CCRBF (Proposed) model, on several computational complexity metrics such as Inference Time (ms), Memory Usage (GB), FLOPs (GFLOPs), and Parameters (Millions). The efficiency of the CCRBF (Proposed) model is shown to be optimal in all aspects, where it has the shortest inference time of 38 ms, the lowest memory usage of 1.4 GB, and the lowest number of parameters of 1.2 million with 5.8 GFLOPs to FLOPs. Overall, the other models, such as Graph-BERT, were observed to have a

Table 7 — Comparison based on computational complexity with existing models

Techniques	Inference time (ms)	Memory usage (GB)	FLOPs	Parameters (Millions)
MatFLDA_RFs[12]	85	2.5	12.8	5.2
EResCNN[13]	95	2.8	13.6	6.1
Graph-BERT[14]	135	3.5	18.7	12.0
MFPIC[16]	65	2.0	10.1	3.7
SSC-PPI[18]	75	2.3	11.5	4.8
DAEPPI[19]	110	3.1	15.3	7.4
CCRBF (Proposed)	38	1.4	5.8	1.2

relatively high computational cost in terms of inference time of 135 ms and memory usage of 3.5 GB, with 12 million parameters.

Ablation experiment

In an attempt to determine the effectiveness of the CGCRBF model in predicting PPI, ablation experiments are conducted. This approach provides insight into the significance of individual components of the CGCRBF model and their contribution to the model's effectiveness.

Table 8 shows the effect of the important components in the CGCRBF model for protein interaction prediction. The CCRBF (Proposed) model achieves better efficiency with the lowest inference time of 38 ms, the lowest memory usage of 1.4 GB, and the lowest parameters of 1.2 million, while still achieving 5.8 GFLOPs in FLOPs. As an example, models such as Graph-BERT have larger computational costs with an inference time of 135 ms, memory usage of 3.5 GB, and 12 million parameters. The above results affirm the proposed model's efficiency in terms of computation while having comparable performance.

The CGCRBF model's efficacy with varying learning rates and batch sizes is presented in Table 9. The GGCRBF model achieved optimal specifications of a learning rate of 1e-03 and a batch size of 32, yielding results of 99.80% accuracy and 99.68% F1-score. The findings further indicate that both a higher learning rate and larger batch sizes lower the performance, especially in terms of precision and sensitivity.

Figure 8 compares the convergence performance of various optimizing algorithms: UPRSO (Proposed), PRSO, Success-Based Optimization Algorithm (SBOA)[30], Dream Optimization Algorithm (DOA)[31], and Cellular Neighbour Optimizer (CNO)[32] in terms of fitness with iterations; the deepening performance of the five algorithm shows that the UPRSO (Proposed) algorithm converges the fastest and most stably, attaining a fitness value quite close to 0 in about 30 iterations. Conversely, the PRSO converges more slowly with a higher fitness value than the other algorithms, while the SBOA, DOA, and CNO more steadily converge fitness values to a minimum, with CNO achieving the lowest fitness near the conclusion of iterations. This illustrates that the UPRSO method performs optimally in optimization problems.

The findings pertaining to the CCRBF model's performance across the four metrics are reflected in Table 10, which summarizes each statistical test.

- P-value: This statistic reveals the degree of statistical significance of differences exhibited in model performance. Since all p-values were less than 0.05, concluded that the performance of the CCRBF model is statistically significant.
- T-value: The t-statistic calculated for each metric shows how much larger or smaller the sample

Table 8 — Ablation Study of the CGCRBF for PPI with UPRSO

Model Variants	Acc (%)	Pre (%)	Sen (%)	Spe (%)	F1-score (%)
Full CGCRBF (with UPRSO)	99.95	99.80	99.98	99.90	99.87
Without Graph Convolution	98.50	98.30	98.40	98.60	98.35
Without Radial Basis Function (RBF)	98.20	97.90	98.10	98.30	98.00
Without Compression (No UPRSO)	97.80	97.50	97.70	97.90	97.60
Without Both Graph Convolution and RBF	95.50	95.10	95.20	95.40	95.10
Random Model (Baseline)	85.50	83.20	85.10	84.80	84.10

Table 9 — Ablation Study on Learning Rate and Batch Size for CGCRBF Model

Learning rate, Batch size	Acc (%)	Pre (%)	Sen (%)	F1-score (%)
0.001, 32	99.80	99.60	99.75	99.68
0.001, 64	99.60	99.50	99.60	99.55
0.01, 32	99.50	99.40	99.50	99.45
0.01, 64	99.20	99.10	99.30	99.15

Table 10 — Statistical Test for CCRBF Model Performance

Metrics	Value	p-value	t-value	F-value	Cohen's d
Accuracy	99.95%	< 0.05	4.50	2.10	1.21
Precision	99.80%	< 0.05	3.80	1.85	1.10
Sensitivity	99.98%	< 0.05	5.20	2.40	1.40
F1-score	99.87%	< 0.05	4.10	2.00	1.15

mean is compared to the mean of the population, and the t-values are higher than thirty to reflect a larger difference.

All of these statistics confirm the findings that the CCRBF model performed significantly better than all other models on every critical metric for PPI prediction.

Discussion

The CCRBF model performs outstandingly on the various PPI datasets, outperforming other models (MFPIC, BiGRU, DAEPPI, and Graph-BERT) in terms of accuracy, precision, sensitivity, and F1-score. The model records an accuracy of 99.95%, precision of 99.80%, sensitivity of 99.98%, and an F1-score of 99.87%, which reflects the reliability and robustness of the model in PPI prediction. The statistical test results confirm the improved performance with p-values of less than 0.05 probabilities through statistical testing in addition to having significant means f-, d-, or t-values reflecting significant differences with the baseline models and among the baseline models. The ablation study revealed that the model's performance was dominated by the important features of graph convolution and radial basis functions, and the optimization steps in UPRSO measure the accuracy of individual placement. All the analyses affirm that the CCRBF model is not only an effective tool but also a predictive model for PPI, and therefore applied in the examination of viral pathogenesis and therapeutic studies.

Limitations of the study

One major disadvantage of the CCRBF model is its reliance on quality protein sequence data. Although it outperforms other models in PPI prediction, its ability is reduced for data that is incomplete, noisy, or of poor quality, which is normally the case in the biological sciences.

Conclusion

This research developed a CCRBF model that represents a substantial step forward in the prediction of PPIs compared to traditional models. This is done by the CCRBF model, which combines GCNs and RBFs to better represent the complex and nonlinear relationships between proteins. Furthermore, pruning, quantization, and knowledge distillation methods are also utilized to improve the efficiency of the model by removing unnecessary parameters, reducing memory requirements, and maintaining high levels of prediction accuracy. The UPRSO is then used to optimize the parameter space dynamically, thus improving the convergence speed and prediction capabilities of the model while successfully avoiding local minima during the optimization process. The inclusion of SHAP also enables the model to have additional interpretability, which enables researchers to understand the degree to which features related to individual protein sequences influenced the predictions of PPIs. Future studies can be focused on applying the proposed framework to other biological datasets, including multi-modal datasets. Further improvements in scalability and optimization methods will enable the model to tackle even more complex biological systems. The inclusion of federated learning methods will also enable the model to be applied in privacy-concerned settings, making it applicable in clinical and drug discovery settings.

Conflict of interest

Both the authors declare no conflict of interest.

References

1 Ozger ZB, A robust protein language model for SARS-COV-2 protein–protein interaction network prediction. *Artif Intell Med*, 142 (2023) 102574

2 Hu J, Dong M, Tang YX & Zhang GJ, Improving protein-protein interaction site prediction using deep residual neural network. *Anal Biochem*, 670 (2023) 115132.

3 Xian L & Wang Y, Advances in computational methods for protein–protein interaction prediction. *Electronics*, 13 (2024) 1059.

4 Hu W & Ohue M, Spatialppi: Three-dimensional space protein-protein interaction prediction with Alphafold Multimer. *Comput Struct Biotechnol J*, 23 (2024) 1214.

5 Idrees S, Paudel KR, Sadaf T & Hansbro PM, Uncovering domain motif interactions using high-throughput protein–protein interaction detection methods. *FEBS Lett*, 598 (2024) 725.

6 Wu L, Tian Y, Huang Y, Li S, Lin H, Chawla N & Li SM, MAPE-PPI: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. (2022).

7 Tahir M, Khan F, Hayat M & Alshehri MD, An effective machine learning-based model for the prediction of protein–protein interaction sites in health systems. *Neural Comput Appl*, 36 (2024) 65.

8 Ino Y, Yamaoka Y, Tanaka K, Miyakawa K, Nishi M, Hatayama Y & Ryo A, Integrated tandem affinity protein purification using the polyhistidine plus extra 4 amino acids (HIP4) tag system. *Proteomics*, 23 (2023).

9 Tugarinov V, Ceccon A & Clore GM, NMR methods for exploring 'dark' states in ligand binding and protein-protein interactions. *Prog Nucl Magn Reson Spectrosc*, 128 (2022) 1.

10 Khan M & Djamei A, Co-immunoprecipitation-based identification of effector–host protein interactions from pathogen-infected plant tissue. *Methods Mol Biol*, (2023).

11 Jha K, Saha S & Singh H, Prediction of protein–protein interaction using graph neural networks. *Sci Rep*, 12 (2022).

12 Li Y, Wang Z, You ZH, Li LP & Hu X, Predicting protein-protein interactions via random ferns with evolutionary matrix representation. *Comput Math Methods Med*, 2022 (2022) 1.

13 Gao H, Chen C, Li S, Wang C, Zhou W & Yu B, Prediction of protein-protein interactions based on ensemble residual convolutional neural network. *Comput Biol Med*, 152 (2023) 106471.

14 Jha K, Karmakar S & Saha S, Graph-bert and language model-based framework for protein–protein interaction identification. *Sci Rep*, 13 (2023).

15 Tran HN, Nguyen PXQ, Guo F & Wang J, Prediction of protein–protein interactions based on integrating deep learning and feature fusion. *Int J Mol Sci*, 25 (2024) 5820.

16 Goktepe YE, Protein-protein interaction prediction using enhanced features with spaced conjoint triad and amino acid pairwise distance. *PeerJ Comput Sci*, 11 (2025).

17 Hoai Nhan T, Quynh NP & Anh Phuong L, Combining handcrafted and learned features using deep learning to improve protein-protein interaction prediction performance. *J Inf Telecommun*, 9 (2025) 151.

18 Ma Z, Min W, Zhang H, Huang Y & Jiang S, SSC-PPI: A subspace structure consistency-based method for protein-protein interactions prediction. *IEEE Trans Comput Biol Bioinform*, (2025) 1.

19 Zhou S, Luo J, Tang M, Li C, Li Y & He W, Predicting protein–protein interactions in microbes associated with cardiovascular diseases using deep denoising autoencoders and evolutionary information. *Front Pharmacol*, 16 (2025).

20 Cao MY, Zainudin S & Daud KM, Feature fusion with attributed deepwalk for protein–protein interaction prediction. *Sci Rep*, 15 (2025).

21 Lan Q, Zheng Z, Tang Z, Qiu X & Yin Z, Protein-protein interaction prediction using bidirectional GRUs with explicit ensemble. *PLoS One*, 20 (2025).

22 Kumar MR, Arulprakasam KR, Kutevska AN, Mutwil M & Thibault G, Yeast knowledge graphs database for exploring *saccharomyces cerevisiae* and *schizosaccharomyces pombe*. *J Mol Biol*, 437 (2025) 169072.

23 Chen Y, Tang Z, Tang Z, Fu L, Liang G, Zhang Y & Wang B, Identification of core immune-related genes CTSK, C3, and IFITM1 for diagnosing helicobacter pylori infection-associated gastric cancer through transcriptomic analysis. *Int J Biol Macromol*, 287 (2025) 138645.

24 Li B, Li X, Li X, Wang L, Lu J & Wang J, Prediction of influenza A virus-human protein-protein interactions using XGBoost with continuous and discontinuous amino acids information. *PeerJ*, 13 (2025).

25 Zhang T, Chen W, Liu Y & Wu L, An intrusion detection method based on stacked sparse autoencoder and improved gaussian mixture model. *Comput Secur*, 128 (2023) 103144.

26 Sathya V, Shakunthala M, Chakravarthy VJ, Radhika K, Rao GN, Jagtap MT & Bani SG, Radial basis convoluted graph neural network based area efficient 1024-point pipelined radix4 FFT processer for ECG heartbeat categorization. *Circuits Syst Signal Process*, (2025).

27 Balaskas K, Karatzas A, Sad C, Siozios K, Anagnostopoulos I, Zervakis G & Henkel J, Hardware-aware DNN compression via diverse pruning and mixed-precision quantization. *IEEE Trans Emerg Top Comput*, 12 (2024) 1079.

28 Kundu R, Chattopadhyay S, Nag S, Navarro MA & Oliva D, Prism refraction search: A novel physics-based metaheuristic algorithm. *J Supercomput*, 80 (2024) 10746.

29 Nohara Y, Matsumoto K, Soejima H & Nakashima N, Explanation of machine learning models using Shapley additive explanation and application for real data in hospital. *Comput Methods Programs Biomed*, 214 (2022) 106584.

30 Lara-Montano OD, Gomez-Castro FI, Gutierrez-Antonio C & Dragoi EN, Success-based optimization algorithm (SBOA): Development and enhancement of a metaheuristic optimizer. *Comput Chem Eng*, 194 (2025) 108987.

31 Lang Y & Gao Y, Dream optimization algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Comput Methods Appl Mech Eng*, 436 (2025) 117718.

32 Yogi B, Roy S, Khan AK, Rawat U, Jangid M & Bhattacharya P, IELTSOC: Enhanced image encryption using combined logistic and Tinkerbell maps with second order cellular automata for internet of things. *Discover Internet Things*, 5 (2025).