Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 1, January 2025, pp. 580~589
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i1.pp580-589

EMSPLA for accurate feature molecular extraction from protein-ligand interactions

Srinidhi Kulkarni V[1,2], Ganesh Dhandapani[2], Kureeckal V. Ramesh[3]
[1]Department of Computer Science and Engineering, Jyothy Institute of Technology, Bangalore, Karnataka, India
[2]Department of Computer Science and Engineering, Jain (Deemed-to-be University), Karnataka, India
[2]Department of Biotechnology, Jain (Deemed-to-be University), Karnataka, India

Article Info

Article history:

Received May 31, 2024
Revised Sep 11, 2024
Accepted Sep 29, 2024

Keywords:

Aligner
CNN
Deep learning
PDBbind
Protein-Ligand

ABSTRACT

Protein-ligand interactions are fundamental in various biological and medical fields, influencing drug discovery and therapeutic development. In recent years, deep learning (DL) has revolutionized the study of these interactions, but significant challenges remain in accurately representing molecular structures for DL models. Traditional featurization techniques often depend on handcrafted features, requiring expert knowledge and potentially missing crucial molecular aspects. This work addresses these challenges by developing and evaluating a novel protein-ligand feature extraction system using an enhanced molecular similarity protein-ligand aligner (EMSPLA). The primary objective is to leverage EMSPLA for similarity matching in protein-ligand interactions, improving predictive model accuracy. The methodology combines convolutional neural networks (CNN) for local feature extraction with an attention module to capture long-distance dependencies, enhancing binding site predictions. Using the PDBbind v.2020 dataset, the EMSPLA model demonstrated superior performance with a root mean square error (RMSE) of 0.67, surpassing current state-of-the-art models. These findings highlight the system’s potential for efficient deployment and scalability, positioning it as a powerful tool in computational biology and drug discovery, ultimately advancing our understanding of protein-ligand interactions.

Corresponding Author:

Srinidhi Kulkarni V
Department of Computer Science and Engineering, Jain (Deemed-to-be University)
Bangalore, Karnataka, India
Email: sudhasuren05@gmail.com

1. INTRODUCTION

A protein-ligand interaction involves the binding of a ligand, which is typically a small molecule, to a specific site on a protein [1]. Proteins are large, complex molecules that play many critical roles in the body, including acting as enzymes, signaling molecules, and structural components [2]. Ligands can be various substances, including drugs, hormones, or neurotransmitters [3]. The interaction between a protein and its ligand is fundamental to many biological processes, as it can alter the protein’s structure and function, often triggering a biological response [4]. Protein-ligand interactions are crucial in various biological and medical fields. They can regulate physiological processes by activating or inhibiting the protein’s function [5]. For instance, in enzymatic reactions, a substrate (ligand) binds to an enzyme (protein) to catalyze a biochemical reaction [6]. In the pharmaceutical industry, understanding these interactions is vital for drug discovery and development. Drugs are designed to target specific proteins, modulating their activity to treat diseases [7]. By binding to their target proteins, these drugs can inhibit or stimulate the protein’s function, thereby altering the disease process. The specificity and affinity of the ligand for its protein target are key factors in determining the efficacy and safety of a drug [8].

In recent years, deep learning (DL) has revolutionized the study of protein-ligand interactions [9]. DL, a subset of machine learning (ML), uses neural networks with many layers (hence "deep") to model complex patterns in data. In the protein-ligand interactions, DL models can predict binding affinities, identify potential binding sites, and even design novel ligands [10]. These models are trained on large datasets of known protein-ligand complexes, learning to recognize subtle patterns that govern binding interactions. For example, convolutional-neural-networks (CNNs) [11] have been used to analyze 3D structures of protein-ligand complexes, capturing spatial features that are critical for binding. Recurrent-neural-networks (RNNs) [12], another type of DL architecture, have been applied to sequential data, such as protein sequences, to predict their interactions with ligands. One significant challenge in leveraging DL for protein-ligand interactions is the need to accurately represent molecular structures in a form that the models can process, this is where featurization comes in [13]. A featurizer converts the complex, multidimensional information of molecular structures into feature vectors, which are numerical representations that capture the essential characteristics of the molecules [14]. Designing an effective featurizer is critical because the quality of the feature vectors directly impacts the performance of the DL models. These features might include information about the molecular geometry, electronic properties, and physicochemical characteristics of the ligand and protein.

Traditional featurization techniques often rely on handcrafted features, which require expert knowledge and may not capture all relevant aspects of the molecules [15]. However, DL models can be used to learn features directly from raw data, a process known as deep representation learning [16]. This approach can uncover more intricate and informative representations that might be missed by manual featurization. As a result, designing a robust featurizer or leveraging DL to automatically extract features is essential for understanding of protein-ligand interactions and improving the accuracy of predictive models in drug discovery and other applications. The contribution of this work is as:

- Designing a DL featurizer model for extraction of molecular target into feature vector.
- Proposing an enhanced molecular similarity protein-ligand aligner (EMSPLA) for enhancing the similarity matching.
- Using CNN to extract local features from neighboring residues from the protein sequences.
- Compared the root mean square error (RMSE) score with current state-of-art works.

This work is organized in the following manner. In section 2, existing DL models of protein-ligand works are discussed. In section 3, the proposed DL approach is discussed. In section 4, the results for proposed EMSPLA are discussed. Finally, in section 5, the conclusion of the work is given.

2. LITERATURE SURVEY

In this section, various protein-ligand approaches for feature extraction are discussed. Wang *et al.* [17], presented a model called multi-channel sub-structure graph-gated recurrent-unit (MCSG-GRU) which consisted of various neural networks having different attention modules which were applied on the molecules sub-structure for predicting and learning different properties of molecules. First, the features from the molecular information were extracted at molecule and node level for getting coarse and fine-grained information. Further, this work utilized bi-directional GRUs for extraction of features for creating a representation of a given molecule. In this work, for evaluation, ESOL, PDBbind, free solvation (FreeSolv), and Lipophilicity datasets were used and RMSE was used for evaluation. Findings showed that the ESOL dataset achieved RMSE of 0.653, Lipophilicity dataset achieved RMSE of 0.653, FreeSolv dataset achieved RMSE of 0.94 and PDBbind dataset achieved RMSE of 1.27. Further, Deng *et al.* [18], presented an architecture which consisted of graph-neural network (GNN) and XGBoost (XGB) called as XGraphBoost for extraction of features from the molecules. The GNN was used as feature extractor where theu proposed three different neural networks, i.e., graph-convolution-network (GCN), directed-message-passing neural network (DMPNN) and gated-graph neural network (GGNN). Also, the GNN was used as predictive model for predicting the molecule characteristics and XGB was used as classification algorithm. Results were evaluated using RMSE for prediction (regression) and area-under-curve receiver-operating-characteristic (AUC-ROC) for classifying the molecules (classification). For evaluation 10 datasets were used where only three datasets, i.e., Lipophilicity, ESOL and FreeSolv were used for prediction (to evaluated the feature extraction). By utilizing the GCN, GGNN, and DMPNN, the RMSE score achieved by the ESOL dataset was 1.470, 1.0236 and 0.329 respectively. Further, for FreeSolv dataset, the GCN, GGNN and DMPNN achieved 3.499, 1.725 and 0.287 respectively. Finally, when evaluating using Lipophilicity, it achieved RMSE of 1.918, 1.005 and 0.453 for GCN, GGNN, and DMPNN respectively.

Li and Jiang [19], proposed a DL architecture called molecular bidirectional-encoder-representations from transformers (Mol-BERT) which combined the representation of efficient molecular structures and was trained utilizing BERT for prediction of molecular characteristics. The BERT model was used as feature extractor and for generating molecules simplified molecular-input line-entry-system (SMILES). Further after extraction from the BERT model the model was fine tuned for prediction. Four datasets were used for evaluation, i.e., ClinTox, Tox21, SIDER and BBBP. The Mol-BERT achieved RMSE of 0.923 for Clintox, 0.839 for Tox 21, 0.875 for BBBP and 0.695 for SIDER. Yang *et al.* [20], presented a GNN having 27 GCN for extracting global and local structures of molecules (i.e., protein-ligand compound), called as molecular-graph drug-target affinity (M-GraphDTA). The also proposed a gradient-weighted affinity-activation-mapping (Grad-AAM) approach for analyzing the chemical properties of molecules. Total of seven datasets were used for evaluation of proposed approaches. The M-GraphDTA achieved an average RMSE of 0.695 for all seven datasets. Wen *et al.* [21], presented a molecular-property predictive (MPP) approach. This work utilized BERT model to extract important features from molecular fingertips, hence was called as fingerprint-BERT (FP-BERT). The features extracted from the FP-BERT were then used as input for the CNN layer for further extracting different features. Evaluation was done using FreeSolc, ESOL, HIV, BBBP, CEP and Malaria datasets. The FP-BERT-CNN achieved RMSE of 1.523 for FreeSolv, 1.22 for Malaria, 1.05 for CEP, 0.602 for Lipophilicity, and 0.552 for ESOL. Further, Gu *et al.* [22], presented a super-edge graph convolution-based supervised attention-based DTA (SEGSA-DTA) which extracted edge and node data from molecules using various attention modules to learn the distribution of protein-ligands interactions. For evaluation, PDBbind was used where SEGSA-DTA achieved RMSE of 1.319.

Xu *et al.* [23], presented an architecture which combined graph-attention-network (GAT) with self-attention long-short term-memory (SA-LSTM) for extraction of features from graph and sequences of molecules. In this work, SMILES were initially used, then GAT was used for feature extraction. The attention layers of GAT and SA-LSTM were both combined to understand the complete molecule. The evaluation was done using various datasets to evaluate their work. The SA-LSTM GAT achieved RMSE of 0.709 for Lipophilicity, 0.885 for ESOL and 1.211 for FreeSolv datasets respectively. Pecina *et al.* [24], proposed a scoring approach called as semi-empirical quantum-mechanical scoring (SQM2.20) to solve the issue of binding. Further, to evaluate its work, the PL-REX dataset which consists of various crystal-like structures having 10 protein targets was considered. The SQM2.20 achieved RMSE of 0.69 for PL-REX and solved the issue of cost in density-functional-theory (DFT). Finally, [25] presented a cleaned version of PDBbind dataset (consisting of protein-ligands) which was free of data leakage and could be used for training and testing. The dataset consisted of long sequences of molecules and having similar structural similarity. The dataset was then used for evaluating the existing scoring models, i.e., random-forst scorin (RFS), interaction-graph-network (IGN), auto-dock vina (ADV) and deep-DTA. They also presented a novel scoring called linear-potential (LP-PDBbind). Findings show that the LP-PDBbind showed better results in comparison with existing approaches.

From the above literature survey, it can be seen that using the FreeSolv, ESOL and Lipophilicity dataset, various models have achieved better results. But still very less work has been considered on PDBBind dataset to achieve less RMSE. Further, it can also be seen that most of the work are utilizing BERT models for feature extraction and similarity scoring approaches for evaluating the similarity among the molecules of protein-ligands and its output given to CNN for better outcomes. Hence, in this work, we present an approach which utilizes BERT model with a novel similarity scoring approach with CNN called as EMSPLA-CNN. The methodology for the EMSPLA-CNN is presented below.

3. METHOD

In this methodology section, we will first discuss the architecture. Next, we will cover the dataset used. Following that, we will explain how the protein-ligands are represented. We will then present the proposed EMSPLA. Finally, we will discuss how the CNN is used for feature extraction from the protein-ligand structural similarity. Then finally, this work discusses the classifier used in this work.

3.1. Architecture

The proposed EMSPLA-CNN architecture first evaluates the given protein-ligand binding structure as given in Figure 1. The evaluation is done by evaluating the protein-ligand sequences. Then the protein-ligand sequences are separated using BERT model for further evaluation using the proposed EMSPLA similarity evaluation. After the evaluation of the similarity, the sequence goes to the CNN where it has two components, a neural-network layer, and attention network. The CNN is used as feature extractor where the local features among the neighbor residues are extracted. Further, for extracting more features from the protein sequences, the CNN was stacked in blocks. Long sequences of protein and their binding residues were captured using the attention network. Finally, outputs from both the network were merged and passed to the final connected layer, where the binary prediction took place using the classifier proposed in this work. The issue of overfitting was prevented by utilizing the dropout and weigh decaying approach (not used in convolution layer). The complete flow of the architecture is presented in Figure 1.

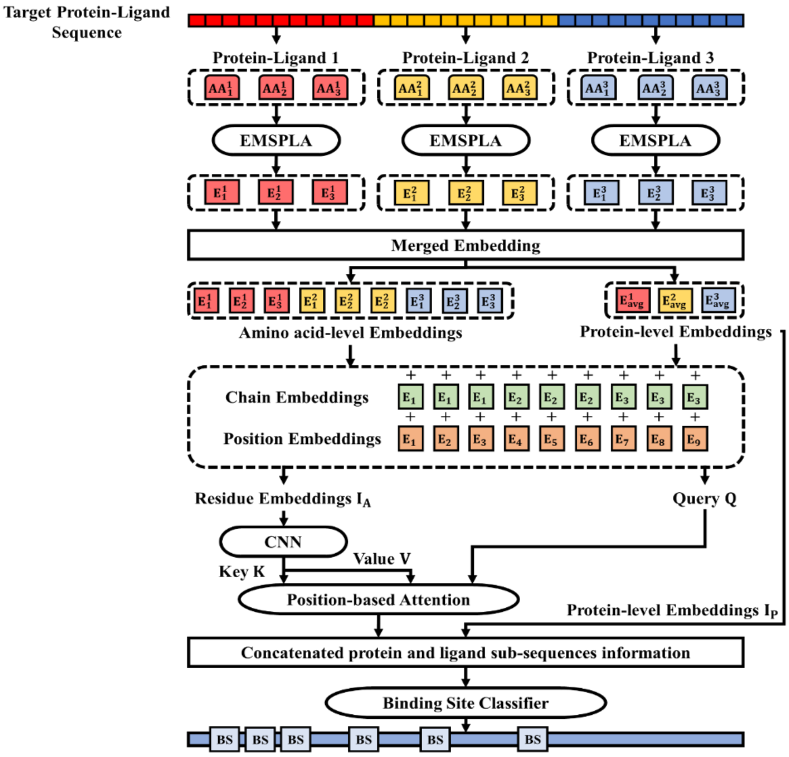

Figure 1. Architecture

3.2. Dataset preprocessing

For evaluating this work the PDBbnind v.2020 [25] dataset was utilized. The PDBbind v.2020 dataset includes binding-site information for 19,443 complexes of protein-ligand that have been identified through experimentation. The information is available in both MOL2 and PDB file formats. The selection of complex protein-ligand from the PDBbind dataset was done by considering different decisions which are discussed as follows. Since the total length of the sequence of protein-ligand differs from one protein-ligand to another and since inclusion of sequences that are too lengthy can degrade EMSPLA-CNN performance, hence, in this work, first imposed a restriction that the protein-ligand sequence could not exceed 1500 amino-acid residues. This led to the exclusion of complex protein-ligand sequences from the datasets when the length of protein-ligand sequence was more than 1500. The second step was the elimination of complexes containing ligands which proved not compatible with the RDKit and Openbabel libraries. After completion of PDBbind preprocessing a total of 34,005 protein-ligand sequences were achieved. Utilizing individual PDB IDs, this work acquired data to train for 28,728 complex protein-ligand sequences. To stop information from leaking out as presented in [26], we separated the test and training datasets and didn't use identical protein-ligand sequences. By comparing the structural similarities of both testing and training protein-ligand utilizing the suggested EMSPLA, this work was able to create unseen-protein test datasets, which we then used to assess binding site predictions for unknown protein-ligand structures.

3.3. Enhanced molecular similarity protein-ligand aligner

According to research [1], a protein-ligand sequence is composed of a sequence of amino-acids arranged in a particular order with ligands concatenating different proteins. It has been determined that there are roughly 20 distinctive amino-acids (standard amino-acids residues), and the characteristics of protein-ligand sequences vary based on the particular sequence of these amino-acids. It has been observed that certain proteins might have synthetic residues (non-standard residues). In order to generate the 21 residues identified in the protein-ligand sequences, this work utilized EMSPLA, a pre-trained approach that is based on the Smith-Waterman (SW) approach specifically designed for analyzing large-scale protein-ligand sequences. Consider that q_0 and q_1 is used for representing two protein-ligand sequences which are identified using BERT model. The main aim of the SW approach is to compute the similarity among the two-protein ligand sequences. The similarity is stored in a matrix form which is called as $Score$ and denoted as Z. Using the $Score$ and back-tracking approach, best alignment is achieved. Further, consider $\mathcal{X}$ which denotes size of q_0 protein-ligand sequence and $\mathcal{Y}$ denotes size of q_1 protein-ligand sequence. It is assumed that for q_0, there exist $\mathcal{X} + 1$ prefixes, Similarly, for q_1, there exists $\mathcal{Y} + 1$ prefixes. Also, the q_0 and q_1 can contain null sequences. Consider $q_{seq}[1 \dots \mathcal{Y}]$ which represents the prefix for $\mathcal{Y}$ and $q_{seq}[\mathcal{X}]$ which represents prefix for $\mathcal{X}$ respectively, where q_{seq} represents the q sequences. Further, the $Score$ among the prefixes $q_0\,[1 \dots a]$ and $q_1[1 \dots b]$ is denoted as $q_{a,b}$. The $Score$ size is denoted as $(\mathcal{X} + 1, \mathcal{Y} + 1)$. The initial column and row of $Score$ is initially set to 0. The $Score$ for the sequence is represented as $Z_{a,b}$ which is evaluated utilizing (1).

$$Z_{a,b} = max \begin{cases} 0 \\ I_{a,b} \\ J_{a,b} \\ Z_{a-1,b-1} - R(a,b) \end{cases} \tag{1}$$

In (1), $R(a,b)$ purpose is to provide results, i.e., mismatch or there is a match between two sequences. If there is a match, then, this is represented as $q_0[a] = q_1[b]$ and $R(a,b) = X_i$. If there is no match, then it is represented as $q_0[a] \neq q_1[b]$ and $R(a,b) = M_i$. Also, in (1), the I and J denote the matrix with respect to the protein-ligand affinity gap approach. The matrix I and J are utilized for evaluating the gaps, hence, are evaluated using (2) and (3) respectively.

$$I_{a,b} = max \begin{cases} I_{a,b-1} - G_E \\ Z_{a,b-1} - G_F \end{cases} \tag{2}$$

$$J_{a,b} = max \begin{cases} J_{a-1,b} - G_E \\ Z_{a-1,b} - G_F \end{cases} \tag{3}$$

In (2) and (3), G_E and G_F denote firs-gap and successive-gap penalty. In this work, the back-racking approach has been used with SW for finding best aligner among q_0 and q_1. The process of back-tracking starts using the $Score$ having the maximum score and goes to the final cell where a null value cell is reached. The purpose of SW approach is to provide best similarity score and best aligners. The computation of $Score$ takes more time as it has to match and perform back-tracking process simultaneously.

Further, in the EMSPLA, for solving the above issues, an indexing approach is used by utilizing query and reference sequence denoted as Q and R. The sequence is for the Q sequence is constructed in the format of directed-acyclic-graph (DAG) which represents the q_0 sequence. For R sequence, a prefix-tree structure is constructed which represents the q_1 sequence. The prefix-tree is constructed in such a manner such that similarity among the edges and nodes can be found in a faster way. In the final process of this approach, a distinctive string is achieved which is a sub-string of R sequence. Further, every node in the given tree is denoted using and interval suffix-array. By using the traversing approach, the nodes which create the string are sorted in lexicographically way. Further, using the prefix-tree, the nodes which are similar or identical to suffix-array are used to create a Prefix DAG (P-DAG). Every node in P-DAG is used to denote single or more sub-string of sequence R. Consider $DG(X)$ and $PT(X)$ are two functions which are used for constructing prefix-tree and P-DAG, where X represents Q or R sequence. In the EMSPLA technique, the $DG(Q)$ and $PT(R)$ are computed initially. Consider a which denotes root-node for $PT(R)$ and b denotes root-node for $DG(Q)$. The best $Score$ among the Q and R sequences are evaluated using dynamic-programming approach. Further, we consider three $Score$, i.e., $E_{a,b}$, $F_{a,b}$, and $K_{a,b}$ which are initially set to null by considering the root-nodes $DG(Q)$ and $PT(R)$. For every parent in tree a_p in $DG(Q)$, evaluation is done for $E_{ab|a_p}$, $F_{ab|a_p}$, and $K_{ab|a_p}$. The evaluation of $F_{ab|a_p}$ is done using (4).

$$F_{ab|a_p} = max\left\{F_{a_p b}, E_{a_p b} - g^o\right\} - g^e \tag{4}$$

Where, g^o represents protein-ligand open-gap penalty and g^e represents the protein-ligand gap-extension function. Further, $K_{ab|a_p}$ is evaluated by utilizing (5). In (5), b_p represents parent for node b in $PT(R)$. Finally, the evaluation of $E_{ab|a_p}$ is done using (6). In (6), $O\left(a_p, a\,;\, b_p, b\right)$ denotes the $Score$ among the edges $\left(a_p, a\right)$ and $\left(b_p, b\right)$. The evaluation of the all the $Scores$, i.e., E_{ab}, F_{ab}, and K_{ab} is done using (7).

$$K_{ab|a_p} = max\left\{K_{ab_p}, E_{ab_p} - g^o\right\} - g^e \tag{5}$$

$$E_{ab|a_p} = max\left\{E_{a_p b_p} + O(a_p, a\,;\, b_p, b), F_{a_p b_p}, K_{a_p b_p}, 0\right\} \tag{6}$$

$$(E_{ab}, F_{ab}, K_{ab}) = \begin{cases} (E_{ab|a'}, F_{ab|a'}, K_{ab|a'}) & When\ E_{ab|a'} > 0 \\ (-\infty, -\infty, -\infty) & all\ other\ cases \end{cases} \tag{7}$$

In (7), $a' = arg \underset{a_p \in\, ar(a)}{max} E_{ab|a_p}$ and $ar(a)$ represent the parent-nodes for a. Further, E_{ab} denotes best $Score$ for sub-string, that is similarity among sub-string a and sub-string b. When $E_{ab} > 0$, then the sub-string b has a good match with sub-string a. As the SW technique provided better aligner but took more time for computation, hence, the EMSPLA reduces time by reducing the computation and by using the traversal approach on $PT(X)$ and $DG(X)$. The dynamic-programming helps to identify better matches among the protein-ligand sequences. Further, a matching-pair (a, b) is constructed when an optimal $Score$ is achieved, that is when E_{ab} is higher and suffix-array is of size b within the threshold value. Further, in the matching process, the optimal match is achieved from (a, b) and by analysis of suffix-arrays of Q and R sequences. Further, the protein-ligands are used for extracting features using the CNN which is discussed in the next section.

3.4. Feature extraction using CNN

Using the protein-ligand structure similarity achieved from EMSPLA, the output is passed on the CNN layer for feature extraction. The EMSPLA scores the protein-ligand in between 0 and 1 indicating that higher score has structure similar to the next protein-ligand sequence. Whenever, the similarity score among two protein-ligand has more than two scores, then the maximum pair-wise chain EMSPLA-score was given as similarity score. Also, the dataset which was considered for testing comprised protein-ligands having $\leq 40\%$ of similarity when compared with the protein-ligand training set.

3.4.1. Embeddings

The EMSPLA-CNN approach further uses BERT approach for extraction of amino-acid embeddings represented as $XA \in R^{L\times d}$ from the protein-ligand sequence as it helps to extract better features when separating the amino acids and proteins from the protein-ligand sequence. The L represent highest sequence length of protein and d represents hidden-size. The sequences which are less in comparison with L are set to zero. Using the amino-acid embeddings XA, the protein embeddings are achieved by averaging amino-acid embeddings and represented as $X_P \in R^d$. In any circumstance where the protein-ligand has multi-chain sequence structure, then is divided into single sequence structure and every amino-acid embedding extraction takes place. Furthermore, the proteins having multi-chain sequence structure, the protein embeddings are achieved in the similar way, i.e., by averaging amino-acid embeddings and is represented using $XP \in R^{N_c\times d}$, where N_c represents total protein-chains. Finally, the position embeddings represented as $E_P \in R^{L\times d}$ are merged with X_A to evaluate the position of every residue. The chain embeddings $E_C \in R^{N_t\times d}$ are further merged with X_A for modelling mutli-chain sequence structure of protein. The N_t represents highest total chains. The residual embedding $I_A \in R^{L\times d}$ are denoted using (8). In (8), $E_c' \in R^{L\times d}$ represents the chain-embedding e_C with respect to chain of every residue.

$$I_A = LayerNorm(X_A + E_P + E_c'\,) \tag{8}$$

3.4.2. Feature extraction

In this work, 1-dimension CNN approach was used for extracting features from the neighboring residues from the protein-ligand sequences. In the training set, 95% of the protein-ligand bindings had ≤ 23 amino-acid residues. Using this, the CNN approach was constructed in such a way that the features could be extracted from the protein-ligand binding sequence. In this work, the CNN had three blocks which were stacked on each layer of CNN, such that features could be extracted in a hierarchal way in the last stacked CNN layer. The stacked layer had different kernel widths as presented in Figure 2, i.e., 7, 5 and 3 with dilation rate of 3, 2 and 1 respectively. In CNN, the convolutional kernel $k \times d \times c$ transformed the $L \times d$ input features into $L \times 1 \times c$ features using (9). In (9), r represents the dilation-rate, c represents total channels and k represents kernel-width.

$$y_c[i] = \sum_{j=1}^{k} x[i + r \cdot j] w_c[j] \tag{9}$$

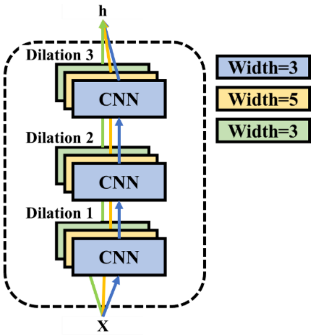

Figure 2. Architecture of proposed CNN model in EMSPLA-CNN

3.4.3. Position-based attention mechanism

The EMSPLA-CNN utilizes the convolution layers for capturing dependency among neighboring protein-ligand binding residues. Nevertheless, by using stacked layers of CNN or by using more kernel-widths, maximum protein-ligand binding dependencies can be extracted. Also, this increases the cost while computation. Hence, to solve this issue, and to capture maximum protein-ligand binding dependencies, this work presents a novel position-based embedding attention approach. The position-based embedding attention helps in extracting maximum features from the protein-ligand binding sequences. Hence, the k which represents key and $V \in R^{L\times d}$ which represents its respective value is obtained from the CNN output layer O_C. Further, the $Q \in R^{L\times d}$ which defines the query is evaluated using (10).

$$Q = LayerNorm(I_p + E_P + E_C) \tag{10}$$

Where, $I_P \in R^{L\times d}$ represents the embeddings which consist of protein-ligand embeddings wit respect to each chain-embedding residue. The relationship among every protein-ligand position and next protein-ligand position is evaluated using (11). The $head_h$ in (11) is evaluated using (12). The attention component in (12) is evaluated using (13).

$$Multihead(Q, K, V) = Concat(head_1, head_2, \dots, head_H)W^O \tag{11}$$

$$head_H = Attention(QW_h^Q, KW_h^K, VW_h^V) \tag{12}$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{13}$$

In (12), W_h^Q belongs to $R^{d\times d_k}$, W_h^K belongs to $R^{d\times d_k}$, W_h^Vbelongs to $R^{d\times d_v}$ and W_h^Q belongs to $R^{d\times d_{model}}$ which are consider as parameter matrix and H represents total heads present in $Multihead$. The final output achieved by the (13) is further used as an input to the feed-forward-network (FFN) which is fully connected. The FFN comprises of two bi-linear transformation having Gaussian-Error Linear-Unit activation function in between of FFN layers.

$$FFN(x) = \sigma(xW_1 + b_1)W_2 + b_2 \tag{14}$$

Further, the normalization and residual relationship are applied to achieve better acceleration for computation and for stable gradients. The final output is achieved as $O_A \in R^{L\times d_{model}}$ which represents the information between the dependencies of protein-ligand sequences.

3.4.4. Classifier

The EMSPLA-CNN uses the final output $O_A \in R^{L\times d_{model}}$ for prediction of the protein-ligand bindings by merging the protein-ligand embeddings I_P and O_A. This is done using (15). The classification block consists of connected layers with 4-hideen layers. Further, the X output is used for final prediction and classification of every position of protein-ligand sequences as binder or non-binder. The output is evaluated using RMSE which is evaluated using (16). In (16), $\hat{y}_i$ are predicted values, y_i are observed values, and n is total observations. Using this proposed EMSPLA-CNN approach, the results were evaluated, which are discussed in the next section.

$$X = I_P \oplus O_A \tag{15}$$

$$FFN(x) = \sigma(xW_1 + b_1)W_2 + b_2 \tag{16}$$

4. RESULTS AND DISCUSSION

The development of the protein-ligand interaction prediction system was accomplished using both Python and C# programming languages, leveraging their respective strengths to create a robust and efficient solution. Python, known for its extensive libraries and frameworks in data processing and ML, played a crucial role in this work. Libraries such as NumPy, Pandas, and SciPy facilitated data manipulation and analysis, while DL frameworks like TensorFlow and PyTorch were utilized for model training and evaluation. Additionally, cheminformatics tools like RDKit were employed to handle molecular data. On the other hand, C# was instrumental in integrating these models into larger software systems, ensuring efficient deployment and scalability through the .NET Core framework. C# libraries such as ML.NET and Accord.NET were used for ML tasks and scientific computations within the application environment. This dual-language approach, combining Python's flexibility and comprehensive ML ecosystem with C#'s performance and integration capabilities, resulted in a powerful system for predicting protein-ligand interactions. The hardware requirements include an Intel Core i7 processor, 16 GB of RAM, and at least 500 GB of free storage, with a dedicated GPU recommended for DL tasks. The software requirements encompass Windows 11, alongside Python version 3.8 or higher and .NET Core 3.1 or higher for C#. This comprehensive setup ensured that the system is both effective and scalable, capable of handling complex computational tasks in the realm of protein-ligand interactions.

The evaluation of the EMSPLA-CNN was done using PDBbind dataset [25]. The PDBbind v.2020 dataset includes binding-site information for 19,443 complexes of protein-ligand that have been identified through experimentation. The information is available in both MOL2 and PDB file formats. The evaluation was done using RMSE score. Also, comparison was done with current state-of-art-works. The results are shown in Figure 3 and in Table 1. The Figure 3 and Table 1 presents a comparison of various models based on their RMSE in predicting protein-ligand interactions, where a lower RMSE indicates better predictive accuracy. Among the models listed, the proposed EMSPLA-CNN stands out with the lowest RMSE of 0.67, suggesting it has the highest predictive accuracy. This model utilizes CNNs to capture complex spatial features of protein-ligand interactions, highlighting the effectiveness of DL techniques. SQM2.20, another recent model from 2024, also demonstrates high accuracy with an RMSE of 0.69, indicating significant improvements over earlier models. In contrast, AutoDockVina, despite being a 2024 model, shows the highest RMSE of 2.85, indicating lower predictive performance which is due to its methodology. Other models like MSGG (2020) and SEGSA-DTA (2023) show moderate performance with RMSEs of 1.27 and 1.319, respectively, while IGN and RF-Score, both from 2024, have slightly higher RMSEs of 1.82 and 1.89. DeepDTA (2024), a DL-based model, performs reasonably well with an RMSE of 1.34 but does not match the accuracy of SQM2.20 and EMSPLA-CNN. These results indicate that newer models, especially those incorporating advanced ML and DL techniques, tend to have better predictive performance. The improvements seen in models like SQM2.20 and EMSPLA-CNN underscore the importance of continued innovation in computational methodologies to enhance the accuracy and reliability of protein-ligand interaction predictions.

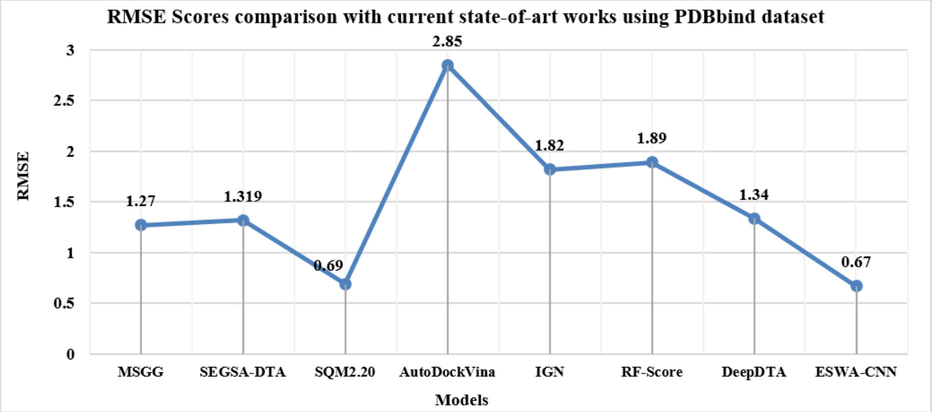

Figure 3. RMSE scores achieved by different models using PDBbind dataset

Table 1. RMSE scores comparison with current state-of-art works and EMSPLA approach

Models, reference, year	RMSE
MSGG, [17], 2020	1.27
SEGSA_DTA, [22], 2023	1.319
SQM2.20, [24], 2024	0.69
AutoDockVina, [26], 2024	2.85
IGN, [26], 2024	1.82
RF-Score, [26], 2024	1.89
DeepDTA, [26], 2024	1.34
EMSPLA-CNN, [Proposed], 2024	0.67

5. CONCLUSION

In this work, this work developed and evaluated the EMSPLA-CNN, a novel protein-ligand feature extraction system that combines CNNs and a position-based attention mechanism. The EMSPLA was used for identifying the similarity between the proteins and CNN was used to extract local features from neighboring residues from the protein sequences. By leveraging the strengths of Python and C#, the system benefited from the extensive ML libraries in Python and the integration and performance capabilities of C#. The EMSPLA approach utilized the PDBbind v.2020 dataset, and through rigorous preprocessing and innovative modeling techniques, the EMSPLA achieved significant improvements in predictive accuracy, as evidenced by an RMSE of 0.67. The EMSPLA-CNN model's ability to capture both local and long-distance dependencies in protein sequences, facilitated by the CNN and attention modules, underscores its robustness and effectiveness. Additionally, the EMSPLA contributes to precise structural similarity assessments, further enhancing the model's predictive performance. The results indicate that integrating advanced DL techniques with traditional sequence alignment methods can substantially improve the accuracy of protein-ligand interaction predictions. This work not only sets a new benchmark in predictive performance but also provides a scalable and efficient framework for future developments in computational biology. Continued innovation in computational methodologies, as demonstrated here, is crucial for advancing the understanding and prediction of protein-ligand interactions. For the future work, the work will be extended for predicting and evaluating the protein-ligand binding affinity.

REFERENCES

[1] K. A. Carpenter and R. B. Altman, "Databases of ligand-binding pockets and protein-ligand interactions," *Computational and Structural Biotechnology Journal*, vol. 23, pp. 1320–1338, Mar. 2024, doi: 10.1016/j.csbj.2024.03.015.

[2] R. Morris, K. A. Black, and E. J. Stollar, "Uncovering protein function: from classification to complexes," *Essays in Biochemistry*, vol. 66, no. 3, pp. 255–285, Aug. 2022, doi: 10.1042/ebc20200108.

[3] R. I. Teleanu, A.-G. Niculescu, E. Roza, O. Vladâcenco, A. M. Grumezescu, and D. M. Teleanu, "Neurotransmitters—key factors in neurological and neurodegenerative disorders of the central nervous system," *International Journal of Molecular Sciences*, vol. 23, no. 11, p. 5954, May 2022, doi: 10.3390/ijms23115954.

[4] H. Lu *et al.*, "Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials," *Signal Transduction and Targeted Therapy*, vol. 5, no. 1, Sep. 2020, doi: 10.1038/s41392-020-00315-3.

[5] X. Du *et al.*, "Insights into protein–ligand interactions: mechanisms, models, and methods," *International Journal of Molecular Sciences*, vol. 17, no. 2, p. 144, Jan. 2016, doi: 10.3390/ijms17020144.

[6] P. Ojeda-May *et al.*, "Dynamic connection between enzymatic catalysis and collective protein motions," *Biochemistry*, vol. 60, no. 28, pp. 2246–2258, Jul. 2021, doi: 10.1021/acs.biochem.1c00221.

[7] K. Dzobo, "The role of natural products as sources of therapeutic agents for innovative drug discovery," *Comprehensive Pharmacology*, pp. 408–422, 2022, doi: 10.1016/B978-0-12-820472-6.00041-4.

[8] R. Friedman, "Computational studies of protein–drug binding affinity changes upon mutations in the drug target," *WIREs Computational Molecular Science*, vol. 12, no. 1, Aug. 2021, doi: 10.1002/wcms.1563.

[9] N. Verma et al., "SSnet: a deep learning approach for protein-ligand interaction prediction," *International Journal of Molecular Sciences*, vol. 22, no. 3, p. 1392, Jan. 2021, doi: 10.3390/ijms22031392.

[10] H. Wang, "Prediction of protein–ligand binding affinity via deep learning models," *Briefings in bioinformatics*, vol. 25, no. 2, Jan. 2024, doi: 10.1093/bib/bbae081.

[11] Y. Wang, Q. Jiao, J. Wang, X. Cai, W. Zhao, and X. Cui, "Prediction of protein-ligand binding affinity with deep learning," *Computational and Structural Biotechnology Journal*, vol. 21, pp. 5796–5806, Jan. 2023, doi: 10.1016/j.csbj.2023.11.009.

[12] E. Elbasani, S. N. Njimbouom, T.-J. Oh, E.-H. Kim, H. Lee, and J.-D. Kim, "GCRNN: graph convolutional recurrent neural network for compound–protein interaction prediction," *BMC bioinformatics*, vol. 22, no. S5, Nov. 2021, doi: 10.1186/s12859-022-04560-x.

[13] A. Dhakal, C. McKay, J. J. Tanner, and J. Cheng, "Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions," *Briefings in Bioinformatics*, vol. 23, no. 1, Nov. 2021, doi: 10.1093/bib/bbab476.

[14] W. Jeon and D. Kim, "FP2VEC: a new molecular featurizer for learning molecular properties," *Bioinformatics*, vol. 35, no. 23, pp. 4979–4985, May 2019, doi: 10.1093/bioinformatics/btz307.

[15] K. Choudhary *et al.*, "Recent advances and applications of deep learning methods in materials science," *npj Computational Material*s, vol. 8, no. 1, Apr. 2022, doi: 10.1038/s41524-022-00734-6.

[16] Y. Si *et al.*, "Deep representation learning of patient data from electronic health records (EHR): a systematic review," *Journal of Biomedical Informatics*, vol. 115, p. 103671, Mar. 2021, doi: 10.1016/j.jbi.2020.103671.

[17] S. Wang, Z. Li, S. Zhang, M. Jiang, X. Wang, and Z. Wei, "Molecular property prediction based on a multichannel substructure graph," *IEEE Access*, vol. 8, pp. 18601–18614, Jan. 2020, doi: 10.1109/access.2020.2968535.

[18] D. Deng, X. Chen, R. Zhang, Z. Lei, X. Wang, and F. Zhou, "XGraphBoost: extracting graph neural network-based features for a better prediction of molecular properties," *Journal of Chemical Information and Modeling*, vol. 61, no. 6, pp. 2697–2705, May 2021, doi: 10.1021/acs.jcim.0c01489.

[19] J. Li and X. Jiang, "Mol-BERT: an effective molecular representation with BERT for molecular property prediction," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–7, Sep. 2021, doi: 10.1155/2021/7181815.

[20] Z. Yang, W. Zhong, L. Zhao, and C. Y.-C. Chen, "MGraphDTA: deep multiscale graph neural network for explainable drug–target binding affinity prediction," *Chemical Science*, vol. 13, no. 3, pp. 816–833, 2022, doi: 10.1039/d1sc05180f.

[21] N. Wen, G. Liu, J. Zhang, R. Zhang, Y. Fu, and X. Han, "A fingerprints based molecular property prediction method using the BERT model," *Journal of Cheminformatics*, vol. 14, no. 1, Oct. 2022, doi: 10.1186/s13321-022-00650-3.

[22] Y. Gu *et al.*, "Protein–ligand binding affinity prediction with edge awareness and supervised attention," *iScience*, vol. 26, no. 1, pp. 105892–105892, Jan. 2023, doi: 10.1016/j.isci.2022.105892.

[23] L. Xu, S. Pan, L. Xia, and Z. Li, "Molecular property prediction by combining LSTM and GAT," *Biomolecules*, vol. 13, no. 3, pp. 503–503, Mar. 2023, doi: 10.3390/biom13030503.

[24] A. Pecina, J. Fanfrlík, M. Lepšík, and J. Řezáč, "SQM2.20: semiempirical quantum-mechanical scoring function yields DFT-quality protein–ligand binding affinity predictions in minutes," *Nature communications*, vol. 15, no. 1, Feb. 2024, doi: 10.1038/s41467-024-45431-8.

[25] Beginner's Guide to the PDBbind Database (v.2020), [Online]. Available: http://www.pdbbind.org.cn/download/pdbbind_2020_intro.pdf

[26] J. Li *et al.*, "Leak proof PDBBind: a reorganized dataset of protein-ligand complexes for more generalizable binding affinity prediction," ArXiv, p. arXiv:2308.09639v2, May 2024, Accessed: May 20, 2024. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/37645037/.

BIOGRAPHIES OF AUTHORS

Srinidhi Kulkarni V obtained his bachelor degree in Information Science and Engineering from Visvesvaraya Technological University Belgaum, Masters in Computer Science and Engineering from Visvesvaraya Technological University Belgaum, Master of Arts in Sanskrit from Karnataka state open University, Diploma in DvaitaVedanta from Tirupathi Sanskrit University. He has 10 years of teaching Experience at undergraduate level of Engineering. He has guided more than 15 Projects and 8 mini projects at undergraduate level. He has numerous research papers in his credit in national and international conference. He has also published a paper in referred journal. He has also participated in several Faculty Development Programs and workshops. His areas of interest include social networks, cloud computing, IoT, big data, design and analysis of algorithms, and data structures. He has the honors of Life Member of ISTE, CSI and is an assistant professor at Jyothy Institute of Technology. He can be contacted at email: sudhasuren05@gmail.com.

Dr. Ganesh Dhandapani has completed his MCA in Bharathiar University Coimbatore, M. Phil at M.S University, Tirunelveli, M.Tech. at Satyabhama University Chennai and Ph.D. in Image processing at Bharathiar University. He has 26+ years of experience in teaching and his professional excellence is through result-oriented approach, hard work, self-motivation and perseverance in teaching carrier. He has published papers in many Scopus journals, conferences and UGC care journals. His areas of research and teaching are image processing, computer networks, cryptography and cloud security, software engineering, and machine learning. He can be contacted at email: ganeshd@gmail.com.

Dr. Kureeckal V. Ramesh is the professor and head of the Department of Biotechnology at JAIN (Deemed-to-be University) in Bengaluru. He specializes in Agricultural Microbiology and Bioinformatics, holding a Ph.D. in Agricultural Microbiology from the University of Agricultural Sciences, Bangalore (2004). With a career spanning over two decades, Dr. Ramesh has held various academic and consulting positions, including roles as Professor and Coordinator, and Lecturer at prominent institutions in Bengaluru. His research focuses on computational biology, particularly in 3D modeling of proteins from microbial sources such as TB bacteria and SARS-CoV-2, docking novel phytochemical ligands on modeled proteins, and in silico studies of the structural and functional diversity of behavioral genes in social organisms like Apis mellifera. He can be contacted at email: kv.ramesh@jainuniversity.ac.in.