J Appl Biotechnol Rep. 2025 September;12(3):1776-1783

Journal of
Applied Biotechnology Reports

doi 10.30491/jabr.2023.405609.1651

Original Article

Assisting *In Silico* Drug Discovery Through Protein-Ligand Binding Affinity Prediction by Convolutional Neural Networks

Milad Rayka [1], **Ali Mohammad Latifi** [1], **Morteza Mirzaei** [1*]

[1] Applied Biotechnology Research Center, New Health Technologies Institute, Baqiyatallah University of Medical Sciences, Tehran, Iran

Corresponding Author: Morteza Mirzaei, PhD, Applied Biotechnology Research Center, New Health Technologies Institute, Baqiyatallah University of Medical Sciences, Tehran, Iran. Tel: +989122366089, E-mail: mraga85@gmail.com

Received November 10, 2023; **Accepted** December 25, 2023; **Online Published** September 30, 2025

Abstract

Introduction: Predicting the binding affinity of ligands and proteins is a vital yet difficult part of structure-based drug design. Recent progress in hardware, particularly GPUs, and the development of efficient deep learning algorithms have significantly increased the application of these technologies to address challenges in drug design. In this study, we introduce a new feature-generation method based on distance-weighted atomic contact, which effectively differentiates between weak and strong interactions. We also examine how the choice of convolutional neural network (CNN) architecture impacts this problem.

Materials and Methods: We used the PDBbind 2016 dataset to train our models. Our approach involved testing different CNN architectures, focusing on a simple, shallow sequential model. The feature-generation method was created to capture key interaction patterns between ligands and proteins. We validated the model's performance using the independent core set of CASF-2016.

Results: Our best model, the Sequential Model, achieved a Pearson's correlation coefficient of 0.79 on the CASF-2016 core set. These results show that a simple, shallow convolutional network paired with a basic feature-generation method can outperform more complex models in this specific case.

Conclusions: This study demonstrates that a vanilla CNN architecture and a simple feature-generation technique can effectively predict ligand-protein binding affinity. The findings indicate that simpler models can deliver highly acceptable performance in structure-based drug design. The Python code needed to reproduce this research is available at https://github.com/miladrayka/convolutional_neural_networks.

Keywords: Deep Learning, Molecular Docking, PDBbind, Scoring Function

Citation: Rayka M, Latifi AM, Ranjbari N, Mirzaei M. Assisting *In Silico* Drug Discovery Through Protein-Ligand Binding Affinity Prediction by Convolutional Neural Networks. J Appl Biotechnol Rep. 2025;12(3):1776-1783. doi:10.30491/jabr.2023.405609.1651

Introduction

Over the past decade, with the advancement of technology, specifically in graphics processing unit (GPU) hardware, the proliferation of data, and the improvement of learning algorithms, deep learning has become an indispensable tool for scientists and engineers.[1,2] Deep learning has a wide range of applications, including face recognition,[3] natural language processing,[4] economics, medicine,[5,6] and drug discovery.[7] Utilizing deep learning for solving challenging chemistry problems has gained numerous attention lately.[8] Deep learning covers various areas of research in chemistry, such as QSAR,[9,10] retrosynthesis,[11] molecular dynamics,[12,13] and quantum chemistry.[14-16] Another crucial area for applying deep learning in drug design is molecular docking.[17]

Identifying novel protein-ligand inhibitors (activators) for every possible protein-ligand pair, either experimentally or by employing molecular dynamics (MD) simulation, is time-consuming and expensive.[18] The molecular docking algorithm circumvents the mentioned obstacles by simplifying the physics of the problem in a two-step process. In step 1, instead of using crystallography techniques or exploring phase space using MD simulation to detect native or quasi-native binding modes, different poses are sampled in the protein's binding site by employing typical algorithms like Monte-Carlo.[19] In step 2, the binding strength of the ligand pose in the protein binding site is estimated using a scoring function, and the pose with the highest score is selected as the most similar structure to the native crystal.[20,21] Estimating this score, which is proportional to binding affinity, is the weakest point in molecular docking. The advent of drug design data sets such as PDBbind[22] and DUD-E[23] contributed to constructing machine and deep learning methods for calculating this demanding task which outperforms all classical scoring functions.[17,24-26]

Feed-forward, convolutional, transformer, and graph neural networks are among the prominent deep learning architectures used for structure-based protein-ligand binding affinity predictions. A feed-forward neural network (FFNN) is composed of several layers consisting of a finite number of neurons.[2] Input data should be converted to a numerical 1-dimensional array. During learning, the parameters of FFNN

Copyright © 2025 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

are optimized so that FFNN can predict the target value of unseen data precisely. NNScore,[27,28] BgN-Score,[29] and Zhu et al.[30] paper are among deep learning-based scoring functions that take advantage of FFNN as a learning algorithm. A convolutional neural network (CNN) is one of the deep learning methods frequently applied in the computer vision field. In contrast to FFNN, CNN's inputs can be two or 3-dimensional grids. Therefore, the 3-dimensional structure of molecules with minimum feature engineering can be used as inputs for training CNN models.[2] Scoring functions in this category have superior performance compare to the other machine and deep learning-based scoring functions. $K_{Deep,}$[31] Pafnucy,[32] OnionNet,[33] LigityScore,[34] Sfcnn,[35] and RosENet[36] are the most well-known scoring functions that employ CNN in their construction. Transformer-based neural networks adopt the mechanism of self-attention, weighing the importance of each part of the input data in their architectures. ViTRMSE, caDeepDock, RTMScore, and HAC-Net[37-40] are some transformer-based examples. Graph neural networks (GNN), or more specifically message passing neural networks (MPNN), have attracted attention in recent years and have become one of the most promising neural networks in the chemistry discipline.[41,42] In MPNN, or GNN each, input data, i.e., molecule, is represented as a graph in which nodes of the graph are atoms, and the bond between them are considered as edges. PotentialNet,[43] SIGN,[44] MetalPro GNet,[45] MedusaGraph,[46] InteractionGraphNet,[47] and graph Delta[48] can be considered MPNN scoring functions.

In this manuscript, three different convolutional neural networks, i.e., sequential, residual, and inception models, and a new feature generation technique based on distance-weighted atomic contacts are proposed to investigate the impact of CNN on binding affinity prediction. The general set of PDBbind 2016 is selected for the training procedure, and the core set is applied as an independent test set. MSE, RMSE, and R_P metrics are reported, in which our best model, the Sequential Model, has the most suitable performance with R_P 0.79 on the core set. Here, we demonstrated that a shallow CNN architecture alongside a simple feature generation technique could provide a suitable estimation of protein-ligand binding affinity. In section 2, we have described the used dataset, feature generation scheme, and CNN architectures. Section 3 is devoted to the results and discussion, and we end the paper with a conclusion in section 4.

Materials and Methods

Here, we adopt the PDBbind 2016 dataset[22] for training and testing our proposed model. We introduce a new feature generation technique that combines GB-Score[50] and OnionNet[33] features to represent each protein-ligand complex. Finally, we explain the CNN architectures, hyperparameters, and training procedures.

Database

PDBbind 2016[22] protein-ligand general set is assigned for training and testing our models. The general set consists of 13308 protein-ligand structures, which we divided into three subsets: train, validation, and test sets. High-quality structures, i.e., core set, were excluded from the general set and used as an independent set, comprising 285 protein-ligand complexes. A validation set of 1,000 protein-ligand complexes was randomly selected from the remaining 13,052 complexes for optimizing hyperparameters and monitoring training procedures. The remaining 12,052 structures were used for the training process.

Defining Descriptors

Every protein-ligand complex should be converted into numerical values so that they can be used during the learning process. For this purpose, the distance-weighted interatomic contacts between ligand and protein atoms in different window ranges are calculated. This type of feature engineering, which is also used in ET-Score[49] and GB-Score,[50] distinguishes between weak and strong interactions and separates them, unlike RF-Score[51] and OnionNet,[33] which only consider contact counts. Ten atom types (H, C, N, O, F, P, S, Cl, Br, I) for ligands and proteins are considered. The process of feature engineering can be summarized in the following formula:

$$x_{i,j,n} = \sum_{k=1}^{K_j} \sum_{l=1}^{L_i} z_{k,l,n}$$

$$z_{k,l,n} = \begin{cases} 1/d_{k,l}, (n-2)\delta + 1 \le d_{k,l} < (n-1)\delta + 1 \\ 0, d_{k,l} < (n-2)\delta + 1, d_{k,l} \ge (n-1)\delta + 1 \end{cases}$$

Where i and j represent ligand and protein atom types, respectively. L_i and K_j are the total numbers of ligand and protein atom types, respectively. Initially, for a predefined atom type, e.g., C of ligand and H of protein, all interatomic distances, $d_{k,l}$ between ligand and protein atoms are calculated. To distinguish between weak and strong interatomic interactions, we applied the inverse of $d_{k,l}$ and assigned them as distance-weighted interatomic contacts. Taking inspiration from the OnionNet[33] scoring function, we constructed 60 shells, n, with 0.5 Å width, δ, from 1 to 30.5 Å. In a given shell, e.g., n = 10, all distance-weighted interatomic contacts between a given atom types whose distances fall within the shell, e.g., 5 and 5.5 Å, are summed together.

This technique separates short- and long-range interatomic interactions and assigns them numerical values based on their strengths and weaknesses. In the end, we derive 6000 features for each complex. By utilizing the power of the convolutional neural network in this work, the

6000 features were transformed into a 2-dimensional grid with dimensions of 60 × 100. The generated features are invariant with respect to translation and rotation, eliminating the need for augmented input data compared to other models that use 3D structures of the complex. In addition to the invariance property, the generated features are unique, compact, and computationally efficient.[52]

Model Architecture and Training

In this work, a family of convolutional neural networks is used for predicting the binding affinity task. Each neural network in this family consists of two parts: convolutional and dense layers. In the dense layer, each neuron is connected to all neurons in the other layers. In contrast, the convolutional layer takes advantage of a mathematical operation called convolution, which reduces the number of parameters (network's weights) and makes it invariant with respect to translation.[2] CNN extracts hierarchical patterns in the input data. For instance, consider a picture. In the lower layers, it can distinguish simple features such as lines and curves. In the upper layers, it can recognize intricate features like eyes or a smile. The input data has a 3D tensor shape, similar to a 2D image with one color channel. Our CNN family encompasses three architectures with distinct characteristics: Sequential, Residual, and Inception neural networks.

The Sequential Model[33] uses three 2D convolutional layers with 32, 64, and 128 filters. Each convolutional layer has a kernel size of 4 and is followed by a 2D max pooling layer with a kernel size of 2 in order to decrease the number of parameters. The output of the last convolutional layer is flattened and used as input for five dense layers with 400, 200, 100, 20, and 1 neurons. Each dense layer is regularized with the L2 penalty with a 0.01 factor, followed by batch normalization and a dropout layer to prevent overfitting. Figure 1S illustrates of the Sequential Model.

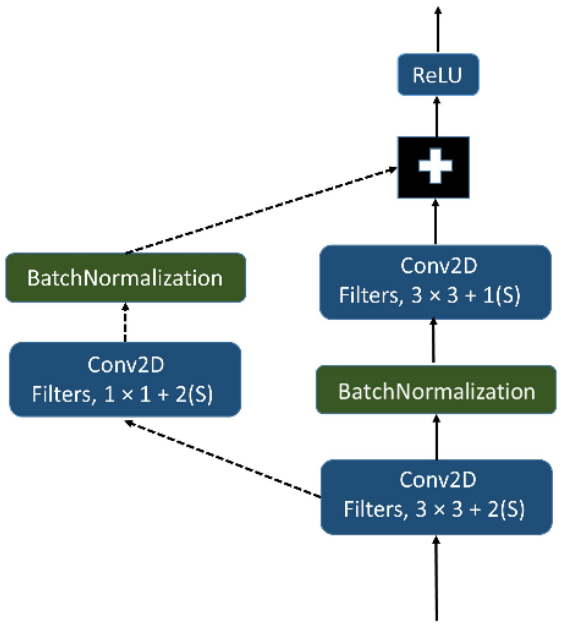

Figure 1. Residual Module. This module includes three 2D convolutional layers and two batch normalization layers with different filters and kernel sizes.

The Residual Model[53] is CNN-based, which takes advantage of skip connections to construct and train a deeper model. The Residual module is illustrated in Figure 1. This module contains three convolutional layers. The convolutional layer in the skip connection (shown by the dotted arrow) has a kernel size of 1 and strides of 2, in contrast to the other layers. Batch normalization is implemented after each layer to prevent vanishing gradients. Finally, the outputs of the two layers are added together and Rectified Linear Units (ReLU) are applied to them. The overall structure of the Residual Model can be imagined as one 2D convolutional layer with 32 filters and a kernel size of 3. This layer is followed by batch normalization and max-pooling layers with a kernel size and strides of 2. Then, five residual modules with 64, 64, 128, 128, and 256 filters are used, making the Residual Model deeper compared to the Sequential Model. In the end, the output of the last layers goes to three dense layers with 100, 50, and 1 neuron. As before, each dense layer is followed by batch normalization and a dropout layer. The detailed structure of the Residual Model is illustrated in Figure 2S.

The Inception model is another CNN-based model which was first employed in the GoogLeNet architecture.[51] By utilizing the Inception module, the model can be deeper and more complicated. This module is represented in Figure 2. The Inception module includes six 2D convolutional layers

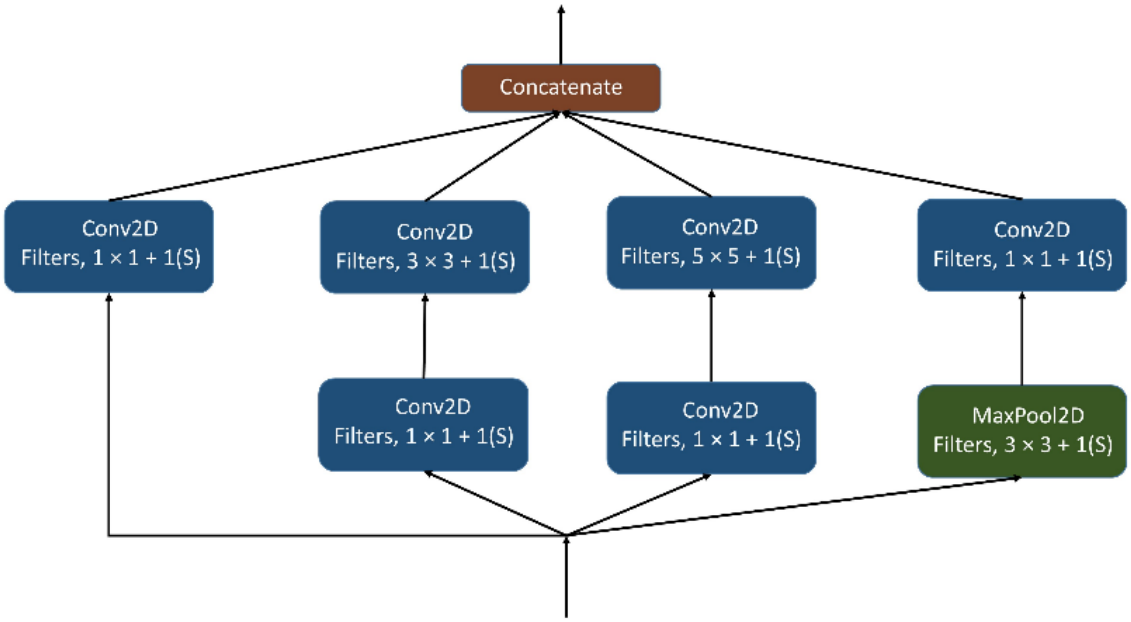

Figure 2. Inception Module. This module includes six 2D convolutional layers and one 2D max pooling layer, each with different filters and kernel sizes but the same strides.

and one 2D max pooling layer, which have distinct filters and kernel sizes but the same strides to extract different kinds of features. A 1 × 1 convolutional layer is used for dimension reduction to reduce computation and the size of the model, therefore, decreasing the probability of the overfitting problem. The model starts with a 2D convolutional layer, which is immediately followed by a 2D max pooling layer to reduce the number of parameters. Then two distinct inception modules are installed to recognize the different patterns in the input data. The output is downsized by a 2D global average pooling layer then the modified data continues to dense layers similar to the Sequential model, which contains batch normalization and dropout layers. Details about the structure and hyperparameters of the Inception Model architecture can be found in Figure 3S.

In all models, the ReLU activation function is applied for both convolutional and dense layers, except for the output layer with one neuron. The probability factor of the dropout layer is fixed at 0.1 for all three models. To optimize the weights of our model during the training procedure, we use the Adam optimizer[52] with a learning rate of 10^{-4}, a batch size of 128, and a mean squared error (MSE) loss function. All models are trained for 500 epochs. To prevent overfitting, we monitor the validation MSE using an early stopping technique.[2] Additionally, mean squared error, root mean squared error (RMSE), and Pearson's correlation (Rp) are reported as metrics for model comparison. Due to the stochastic nature of the training procedure, each model is trained five times, and the mean and standard deviation of their metrics are reported. All models are implemented using Keras and Tensorflow[53] as the backend.

Results and Discussion

A suitable scoring function should perform well on the four tasks.[22] The first task, scoring power, assesses how well a scoring function can generate binding scores that correlate linearly with experimental measurements. The second task, ranking power, evaluates the scoring function's ability to accurately rank known ligands of a specific target protein based on their binding affinities. The final two tasks, docking power, and screening power, focus on evaluating the scoring function's ability to identify the correct native ligand pose and true binders of a target protein, respectively. For this study, our priority is evaluating the scoring power, and we will postpone the investigation of the other tasks to future studies.

Table 1. Calculated Metrics for both the Validation and Core Set of PDBbind 2016. All models were trained five times, and the mean and standard deviation for each metric are shown

Data set	Validation set			Core set		
Model \ Metric	MSE	RMSE	R_P	MSE	RMSE	R_P
Sequential	1.679(0.060)	1.295(0.023)	0.727(0.012)	1.830(0.032)	1.353(0.012)	0.790(0.005)
Residual	1.505(0431)	1.318(0.036)	0.713(0.015)	1.993(0.079)	1.410(0.027)	0.768(0.011)
Inception	1.868(0.081)	1.366(0.030)	0.683(0.016)	2.165(0.094)	1.471(0.031)	0.762(0.013)

Figure 3. Scatter and Histogram Plots of Predicted vs. Experimental Values for the Core Set of PDBbind 2016 Using the Sequential Model.

Three distinct models, i.e., Sequential, Residual, and Inception, were trained on the PDBbind 2016 dataset and tested on the core set in which MSE, RMSE, and R_P metrics were reported for comparison. The results of the experiments for both the core and validation sets are depicted in Table 1. All models achieved an R_P of more than 0.7, indicating their ability to appropriately describe trends in the data. However, the Sequential Model outperformed the other models with an R_P of 0.79 and an RMSE of 1.353. The main difference between these models lies in their convolutional layers. This result confirms that simple and shallow convolutional layers, in this specific case, perform better compared to intricate convolutional layers such as inception and residual modules.

Scatter and histogram plots of predicted binding affinity of the protein-ligand complex with respect to experimental values of the core set data for the Sequential Model, as a superior model, are jointly represented in Figure 3. Although Figure 3 demonstrates that the Sequential Model can generate binding scores that correlate linearly with experimental measurements, it cannot precisely predict the binding affinity values for data with pK_d of more than 10. In our opinion, this defect is related to the shortage of data with pKd of more than ten in the selected training set. By adding appropriate training data or generating more meaningful features that capture subtle chemical information efficiently, this defect can be resolved.

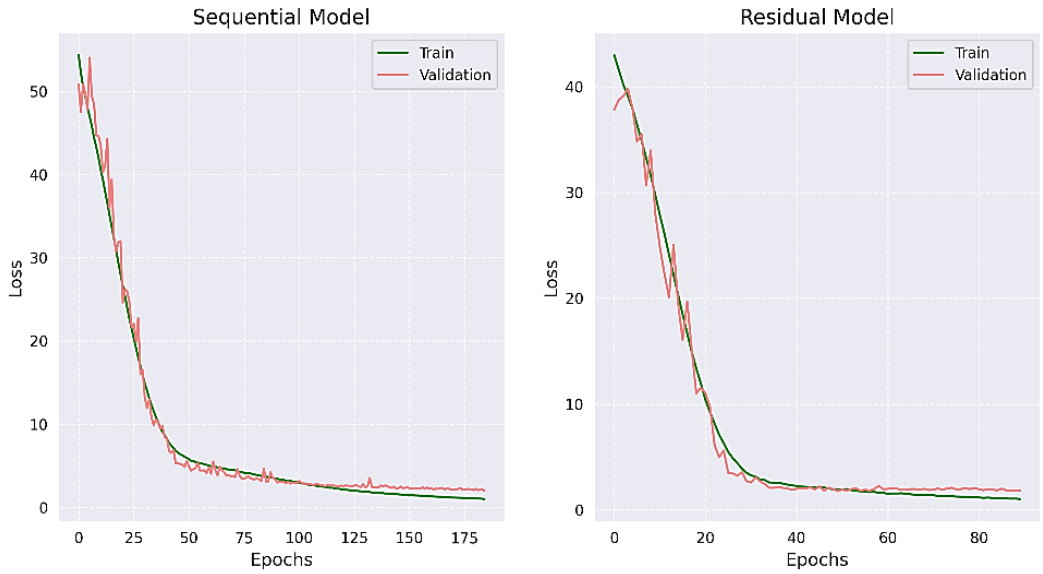

Figure 4. Training and Validation Histories of each Epoch for Sequential and Residual Models.

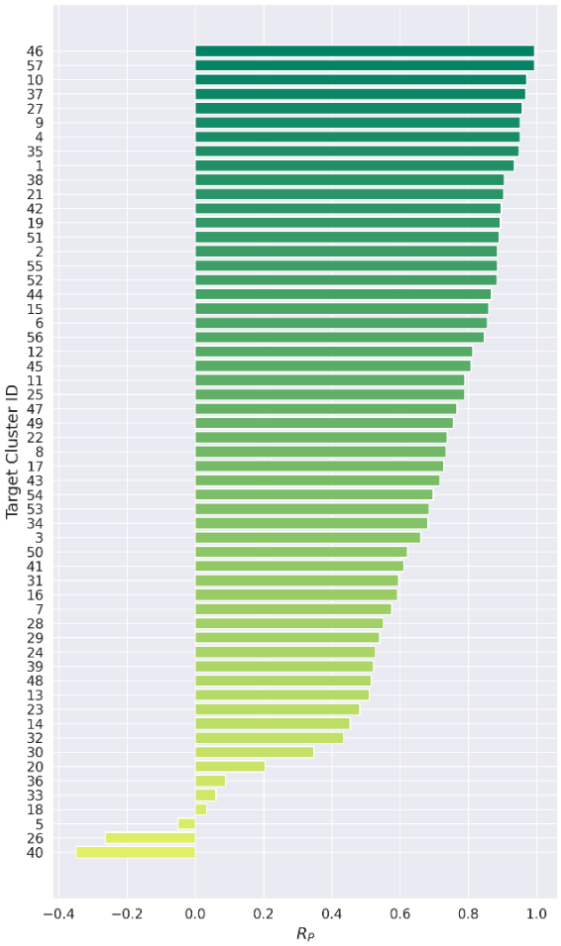

Figure 5. Disaggregating R_P per Target Cluster of Core Set.

Training and validation histories of each epoch for Sequential and Residual models are illustrated in Figure 4. This figure clearly shows that applying dropout, L2 regularization, and early stopping prevents our models from the overfitting problem, allowing them to generalize better to unseen test sets. Figure 4S displays the same graph for the Inception Model.

The core set, or the test set, which includes 285 protein-ligand complexes, consists of 57 protein or target families. Disaggregating RP and RMSE per target cluster of the core set is represented in Figures 5 and 5S, which implies the Sequential Model can accurately predict binding affinity values for a diverse set of protein-ligand complexes. Target clusters with 46 and 40 IDs have the most and the least RP values, respectively. Target cluster 46 belongs to the MTA/SAH nucleosidase protein family, which consists of 5 complexes in the general set. In contrast, target cluster 40 belongs to the Androgen receptor protein family, which includes 27 complexes in the general set. Although the Androgen receptor has more related complexes in the training set, the Sequential Model cannot capture chemical patterns in the MTA/SAH nucleosidase family.

Table 2 provides a comparison of the scoring power (based on the core set 2016) between our proposed model, the Sequential Model, and other suggested CNN-based scoring functions, as well as one traditional scoring function. It is worth mentioning that the scoring functions in Table 2 incorporate varied training sets to train deep learning algorithms, which makes the comparison ambiguous and unfair.

Table 2. A comparison between the Sequential Model, three CNN-based scoring functions (Pafnucy, K_{Deep}, and OnioNet) and one traditional scoring function (X-Score)

Name	X-Score	Pafnucy	K_{Deep}	OnionNet	Sequential Model
Performance (R_P)	0.631	0.78	0.82	0.816	0.79

K_{Deep},[31] Pafnucy,[32] and OnionNet[33] are among the popular scoring functions that apply CNN to predict protein-ligand binding affinity. K_{Deep} utilizes a voxelized representation of the binding site, considering eight pharmacophoric-like properties for featurizing the complex. Pafnucy, on the other hand, employs a different featurization technique for voxelized representation. OnionNet uses element pair-specific contacts between ligands and protein atoms, similar to RF-Score.[51] However, it categorizes these contacts into different distance ranges and creates a grid-shaped input for its CNN. X-Score is a knowledge-based scoring function, which is selected as an example of traditional scoring functions.

Table 2 shows that deep learning-based scoring functions have better performance compared to X-Score. This improvement can be attributed to the non-linear behavior of deep learning algorithms. Among the CNN-based models, KDeep performs the best. However, it is worth mentioning that utilizing a voxelized representation of the protein-ligand complex in KDeep increases the computational cost of binding affinity prediction, which may hinder its application for large datasets. The Sequential Model ranks third among scoring functions, with performance differences typically seen in the second decimal place.

Conclusion

Binding affinity prediction is one of the most crucial steps in structure-based drug design. With the advent of sophisticated deep learning methods, numerous deep learning-based scoring functions have been devised, which outperform classical scoring functions in different tasks. In this manuscript, we investigated the impact of convolutional layers on binding affinity predictions by training three distinct models: Sequential, Residual, and Inception, on the PDBbind 2016 general set and then testing them on the core

set as an independent set. We introduced distance-weighted atomic contact features that distinguish between weak, strong, near, and far interactions. Our results demonstrate that the Sequential Model, despite its more simplistic architecture and feature generation technique, has superior performance compared to the Residual and Inception models with an R_P of 0.79. In the future, we aim to generalize the Sequential Model and our previous scoring function to more complex problems, such as protein-protein and drug-protein interactions, and investigate their abilities in docking and virtual screening tasks.

Authors' Contributions

MR developed the theory, performed the computations, and wrote the main manuscript text. AM. and MM verified the results, discussions, and contributed to the final manuscript. MM supervised the findings of this work. All authors approved the final version of the manuscript.

Data Availability

All codes for repeating this paper are provided in the following GitHub address: https://github.com/miladrayka/convolutional_neural_networks. PDBbind 2016 dataset is accessible free of charge at: http://www.pdbbind.org.cn/.

Conflict of Interest Disclosures

The authors declare that they have no conflicts of interest.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436-44. doi:10.1038/nature14539
2. Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. Cambridge: MIT press; 2016.
3. Parkhi O, Vedaldi A, Zisserman A. Deep face recognition. BMVC 2015-Proceedings of the British Machine Vision Conference 2015.
4. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. arXiv. arXiv preprint arXiv:2005.14165. 2020.
5. Zheng S, Trott A, Srinivasa S, Naik N, Gruesbeck M, Parkes DC, et al. The ai economist: Improving equality and productivity with ai-driven tax policies. arXiv preprint arXiv:2004.13332. 2020.
6. Topol E. Deep medicine: how artificial intelligence can make healthcare human again. Hachette UK; 2019.
7. Stokes JM, Yang K, Swanson K, Jin W, Cubillos-Ruiz A, Donghia NM, et al. A deep learning approach to antibiotic discovery. Cell. 2020;180(4):688-702. doi:10.1016/j.cell.2020.01.021
8. Mater AC, Coote ML. Deep learning in chemistry. J Chem Inf Model. 2019;59(6):2545-59. doi:10.1021/acs.jcim.9b00266
9. Zheng S, Yan X, Yang Y, Xu J. Identifying structure–property relationships through SMILES syntax analysis with self-attention mechanism. J Chem Inf Model. 2019;59(2):914-23. doi:10.1021/acs.jcim.8b00803
10. Yang K, Swanson K, Jin W, Coley C, Eiden P, Gao H, et al. Analyzing learned molecular representations for property prediction. . J Chem Inf Model. 2019;59(8):3370-88. doi:10.1021/acs.jcim.9b00237
11. Zheng S, Rao J, Zhang Z, Xu J, Yang Y. Predicting retrosynthetic reactions using self-corrected transformer neural networks. J Chem Inf Model. 2019;60(1):47-55. doi:10.1021/acs.jcim.9b00949
12. Zhang L, Han J, Wang H, Car R, E W. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. Phys Rev Lett. 2018;120(14):143001. doi:10.1103/PhysRevLett.120.143001
13. Shen L, Yang W. Molecular dynamics simulations with quantum mechanics/molecular mechanics and adaptive neural networks. J Chem Theory Comput. 2018;14(3):1442-55. doi:10.1021/acs.jctc.7b01195
14. Schütt KT, Sauceda HE, Kindermans PJ, Tkatchenko A, Müller KR. Schnet–a deep learning architecture for molecules and materials. J Chem Phys. 2018;148(24):241722. doi:10.1063/1.5019779
15. Unke OT, Meuwly M. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. J Chem Theory Comput. 2019;15(6):3678-93. doi:10.1021/acs.jctc.9b00181
16. Gao X, Ramezanghorbani F, Isayev O, Smith JS, Roitberg AE. TorchANI: a free and open source PyTorch-based deep learning implementation of the ANI neural network potentials. J Chem Inf Model. 2020;60(7):3408-15. doi:10.1021/acs.jcim.0c00451
17. Yang C, Chen EA, Zhang Y. Protein–ligand docking in the machine-learning era. Molecules. 2022;27(14):4568. doi:10.3390/molecules27144568
18. Varela-Rial A, Majewski M, De Fabritiis G. Structure based virtual screening: Fast and slow. Wiley Interdisciplinary Reviews: Comput Mol Sci. 2022;12(2):e1544. doi:10.1002/wcms.1544
19. Koes DR, Baumgartner MP, Camacho CJ. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. J Chem Inf Model. 2013;53(8):1893-904. doi:10.1021/ci300604z
20. Bender BJ, Gahbauer S, Luttens A, Lyu J, Webb CM, Stein RM, et al. A practical guide to large-scale docking. Nat Protoc. 2021;16(10):4799-832. doi:10.1038/s41596-021-00597-z
21. Waszkowycz B, Clark DE, Gancia E. Outstanding challenges in protein–ligand docking and structure-based virtual screening. Wiley Interdiscip Rev Comput Mol Sci. 2011;1(2):229-59. doi:10.1002/wcms.18
22. Su M, Yang Q, Du Y, Feng G, Liu Z, Li Y, et al. Comparative assessment of scoring functions: the CASF-2016 update. J Chem Inf Model. 2018;59(2):895-913. doi:10.1021/acs.jcim.8b00545
23. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. J Med Chem. 2012;55(14):6582-94. doi:10.1021/jm300687e
24. Shen C, Ding J, Wang Z, Cao D, Ding X, Hou T. From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. Wiley Interdiscip Rev: Comput Mol Sci. 2020;10(1):e1429. doi:10.1002/wcms.1429
25. Li H, Sze KH, Lu G, Ballester PJ. Machine-learning scoring functions for structure-based drug lead optimization. Wiley Interdiscip Rev: Comput Mol Sci. 2020;10(5):e1465. doi:10.1002/wcms.1465
26. Sánchez-Cruz N. Deep graph learning in molecular docking: Advances and opportunities. Artif Intell Life Sci. 2023;3:100062. doi:10.1016/j.ailsci.2023.100062
27. Durrant JD, McCammon JA. NNScore: a neural-network-based scoring function for the characterization of protein− ligand complexes. J Chem Inf Model. 2010;50(10):1865-71. doi:10.1021/ci100244v
28. Durrant JD, McCammon JA. NNScore 2.0: a neural-

network receptor–ligand scoring function. J Chem Inf Model. 2011;51(11):2897-903. doi:10.1021/ci2003889
29. Ashtawy HM, Mahapatra NR. BgN-Score and BsN-Score: bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein-ligand complexes. BMC bioinformatics. 2015;16(Suppl 4):S8. doi:10.1186/1471-2105-16-S4-S8
30. Zhu F, Zhang X, Allen JE, Jones D, Lightstone FC. Binding affinity prediction by pairwise function based on neural network. J Chem Inf Model. 2020;60(6):2766-72. doi:10.1021/acs.jcim.0c00026
31. Jiménez J, Skalic M, Martinez-Rosell G, De Fabritiis G. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. J Chem Inf Model. 2018;58(2):287-96. doi:10.1021/acs.jcim.7b00650
32. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. Bioinformatics. 2018;34(21):3666-74. doi:10.1093/bioinformatics/bty374
33. Zheng L, Fan J, Mu Y. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. ACS Omega. 2019;4(14):15956-65. doi:10.1021/acsomega.9b01997
34. Azzopardi J, Ebejer JP. LigityScore: A CNN-Based Method for Binding Affinity Predictions. InInternational Joint Conference on Biomedical Engineering Systems and Technologies. Cham: Springer International Publishing. 2021. pp. 18-44. doi:10.1007/978-3-031-20664-1_2
35. Wang Y, Wei Z, Xi L. Sfcnn: a novel scoring function based on 3D convolutional neural network for accurate and stable protein–ligand affinity prediction. BMC Bioinformatics. 2022;23(1):222. doi:10.1186/s12859-022-04762-3
36. Hassan-Harrirou H, Zhang C, Lemmin T. RosENet: improving binding affinity prediction by leveraging molecular mechanics energies with an ensemble of 3D convolutional neural networks. J Chem Inf Model. 2020;60(6):2791-802. doi:10.1021/acs.jcim.0c00075
37. Guo L, Wang J. ViTRMSE: a three-dimensional RMSE scoring method for protein-ligand docking models based on Vision Transformer. 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2022. pp. 328-333. doi:10.1109/BIBM55620.2022.9995694
38. Wang J, Qiu Z, Zhang X, Yang Z, Zhao W, Cui X. Boosting deep learning-based docking with cross-attention and centrality embedding. 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2022. pp. 360-365. doi:10.1109/BIBM55620.2022.9995214
39. Shen C, Zhang X, Deng Y, Gao J, Wang D, Xu L, et al. Boosting protein–ligand binding pose prediction and virtual screening based on residue–atom distance likelihood potential and graph transformer. J Med Chem. 2022;65(15):10691-706. doi:10.1021/acs.jmedchem.2c00991
40. Kyro GW, Brent RI, Batista VS. Hac-net: A hybrid attention-based convolutional neural network for highly accurate protein–ligand binding affinity prediction. J Chem Inf Model. 2023;63(7):1947-60. doi:10.1021/acs.jcim.3c00251
41. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Networks Learn Syst. 2020;32(1):4-24. doi:10.1109/TNNLS.2020.2978386
42. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. International conference on machine learning. PMLR. 2017. pp. 1263-1272.
43. Feinberg EN, Sur D, Wu Z, Husic BE, Mai H, Li Y, Sun S, Yang J, Ramsundar B, Pande VS. PotentialNet for molecular property prediction. ACS Cent Sci. 2018;4(11):1520-30. doi:10.1021/acscentsci.8b00507
44. Li S, Zhou J, Xu T, Huang L, Wang F, Xiong H, et al. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021. pp. 975-985. doi:10.1145/3447548.3467311
45. Jiang D, Ye Z, Hsieh CY, Yang Z, Zhang X, Kang Y, et al. MetalProGNet: a structure-based deep graph model for metalloprotein–ligand interaction predictions. Chem Sci. 2023;14(8):2054-69. doi:10.1039/D2SC06576B
46. Jiang H, Wang J, Cong W, Huang Y, Ramezani M, Sarma A, et al. Predicting protein–ligand docking structure with graph neural network. J Chem Inf Model. 2022;62(12):2923-32. doi:10.1021/acs.jcim.2c00127
47. Jiang D, Hsieh CY, Wu Z, Kang Y, Wang J, Wang E, et al. InteractionGraphNet: a novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. J Med Chem. 2021;64(24):18209-32. doi:10.1021/acs.jmedchem.1c01830
48. Karlov DS, Sosnin S, Fedorov MV, Popov P. graphDelta: MPNN scoring function for the affinity prediction of protein–ligand complexes. ACS Omega. 2020;5(10):5150-9. doi:10.1021/acsomega.9b04162
49. Rayka M, Karimi-Jafari MH, Firouzi R. ET-score: Improving Protein-ligand Binding Affinity Prediction Based on Distance-weighted Interatomic Contact Features Using Extremely Randomized Trees Algorithm. Mol Inform. 2021;40(8):2060084. doi:10.1002/minf.202060084
50. Rayka M, Firouzi R. GB-score: Minimally designed machine learning scoring function based on distance-weighted interatomic contact features. Mol Inform. 2023;42(3):2200135. doi:10.1002/minf.202200135
51. Ballester PJ, Mitchell JB. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. Bioinformatics. 2010;26(9):1169-75. doi:10.1093/bioinformatics/btq112
52. Himanen L, Jäger MO, Morooka EV, Canova FF, Ranawat YS, Gao DZ, et al. DScribe: Library of descriptors for machine learning in materials science. Comput Phys Commun. 2020;247:106949. doi:10.1016/j.cpc.2019.106949
53. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition 2015. pp. 1-9.