INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 10, Issue 5 - V10I5-1386)
Available online at: https://www.ijariit.com

AFBAP: Attention For Binding Affinity Prediction

Harihar Prasad
harihar052651@greenwoodhigh.edu.in
Greenwood High International School, Bengaluru

ABSTRACT

This paper introduces the AFBAP model, a novel machine learning model that leverages transfer learning benefits from pre-trained transformers ProtBert and ChemBERTa for feature extraction and utilises a CNN-based prediction module prefaced by a task-adaptive feature transformation to predict protein-ligand binding affinity with state-of-the-art accuracy. It accepts one-dimensional sequential inputs for both proteins and ligands, in the form of amino acid strings and SMILES strings respectively. AFBAP's performance over a number of datasets using standard evaluation metrics validates the fact that the model achieves higher accuracy with lower training times and lower compute. AFBAP democratizes access to computational methods of optimizing drug discovery, paving the way for rapid and accessible innovation in drug discovery research.

Keywords: *Protein-ligand binding affinity, Transformers, Convolutional Neural Network, Task-adaptive feature transformations, Transfer Learning, Binding Affinity, Drug Discovery, Machine Learning, BERT*

1.INTRODUCTION

Protein-ligand interactions lay the foundations for understanding a wide range of biological processes, from DNA transcription regulation to immune response. Drugs function as ligands when bonding with bacteria proteins to exercise their remedial functions, and thus understanding protein-ligand interactions are essential to improving drug development [1]. The strength of the interaction between ligands and proteins are characterized by their *binding affinity* (K_d), and accurately determining this value is critical in drug screening, optimization, and repurposing efforts [2].

Traditionally, experimental methods were considered the most reliable method of determining protein-ligand binding affinity, but they only provide results for a very limited range of protein-ligand interactions and utilise supersized amounts of resources and time [3]. In order to determine binding affinity more effectively, in silico computational methods were developed, allowing scientists to determine a wider range of protein-ligand interactions faster and more economically, streamlining the drug discovery process [4].

The first wave of computational models were predominantly physics-based – molecular dynamics simulations, to be exact [5,6,7]. These models were not very accurate, with faulty scoring functions and fundamentally flawed protein and ligand representations undermining their effectiveness [8]. Despite their failure, these models were the precursor to the machine learning and deep learning models used today and helped us understand the underlying theory behind protein-ligand interactions [9].

Of the new models, deep learning models hold the most promise, as they excel at deriving patterns and high-level features from initial data [10]. These models can be divided into two main types: structure-based, and sequence-based models. Structure based models utilize three-dimensional representations of the protein and ligand, while sequence-based models accept one-dimensional string representations as input while achieving comparable accuracy [11, 12, 17, 18, 19].

Transformers are a relatively new architecture that excel at Natural Language Processing (NLP) tasks, such as tokenization and pattern recognition [16]. Through these attributes, they proved incredibly effective at predicting protein-ligand binding affinities from one-dimensional inputs [20]. Their effectiveness at such tasks were recognised and leveraged in models such as CAPLA, which used a cross-attention mechanism. However, such an approach is extremely computationally intensive, and inaccessible to most traditional researchers, forgoing the benefits of transfer learning [21, 14]. Blanchard et. al's model for developing drugs to combat SaRS-COV-2 utilised the benefits of pre-trained encoders extensively and displayed impressive generalization capabilities, but similar performance could be achieved with a lot less computational resources and a series of simpler models.

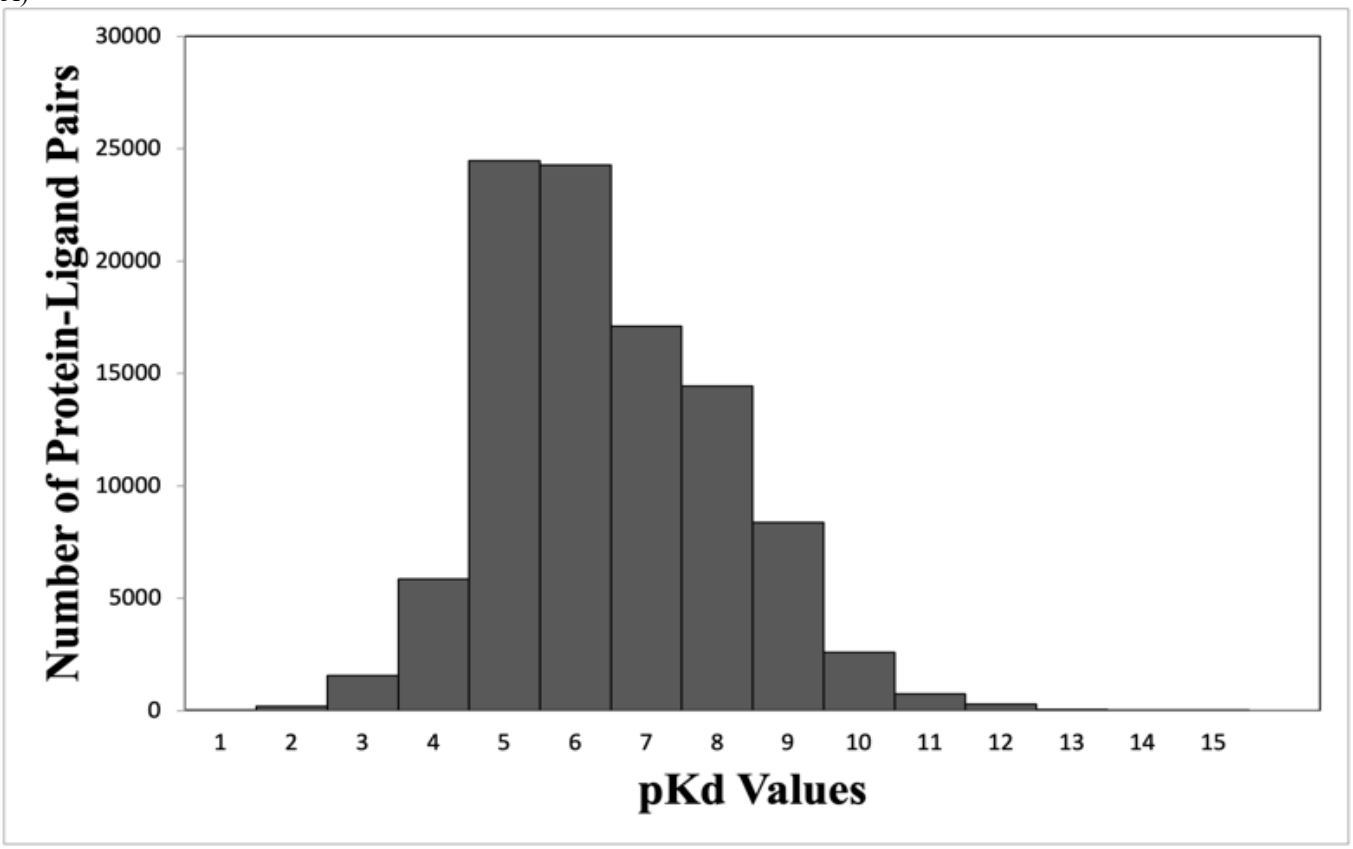
This paper puts forth the Attention For Binding Affinity Prediction (AFBAP) model, which efficiently utilises pre-trained transformer encoders like ProtBert and ChemBerta [23, 19], leveraging transfer learning benefits to achieve state-of-the-art accuracy with minimal computational resources.

AFBAP performs feature extraction using these pre-trained models, whose feature embeddings are then fed into the innovative Adaptive Fully-Connected Layer architecture that performs a task-adaptive feature transformation to increase focus on the relevant feature dimensions, and then predicts binding affinities with a high degree of accuracy. Taking inspiration from Blanchard et. al's model for SARS-CoV-2 inhibitor prediction, AFBAP eschews explicit protein pocket information, instead taking advantage of the ProtBert encoder's representational capacity and considering full amino acid sequences to simplify data acquisition. Our performance metric analysis demonstrates AFBAP's superior accuracy compared to other protein-ligand prediction models, while its utilisation of pre-trained open source transformers reduces computation and data requirements by a large margin. This efficiency not only showcases that state-of-the-art results can be achieved with less compute and data, but also paves the way for rapid and accessible innovation in drug discovery research.

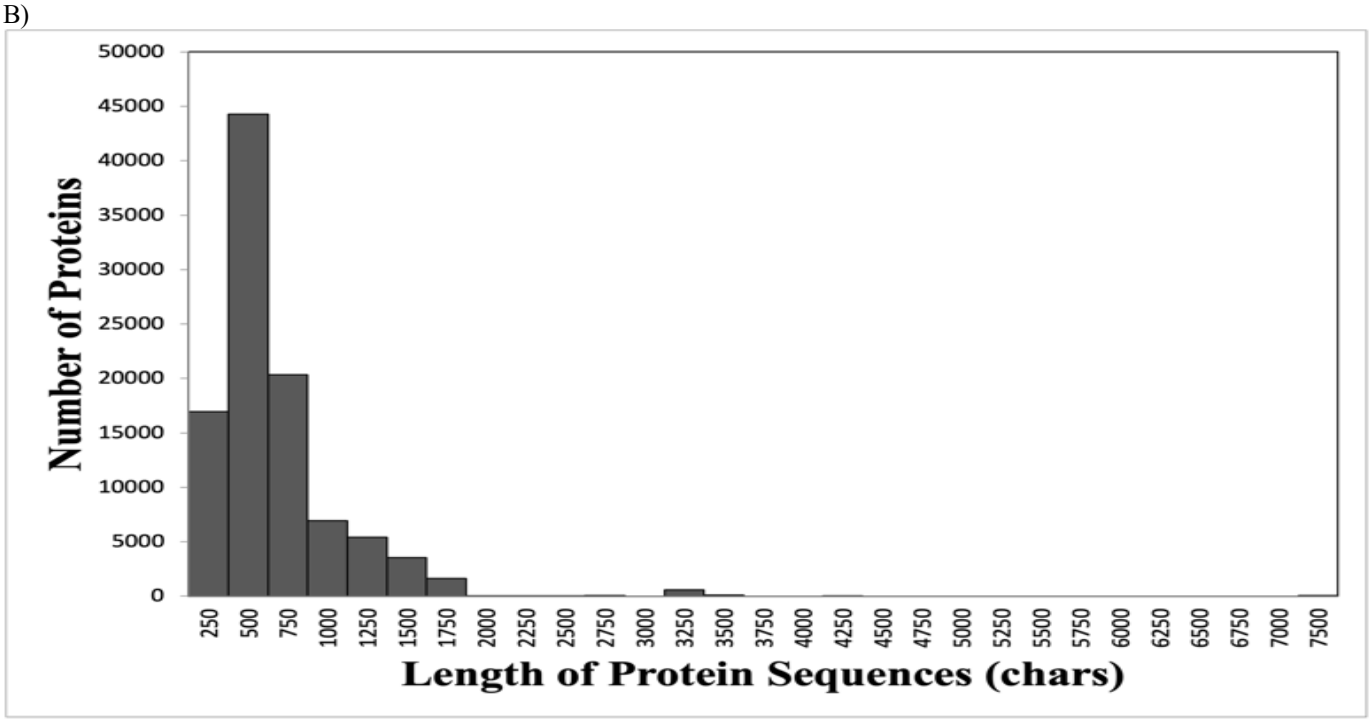
2.METHODS

2.1 Dataset

A)



B)



C)

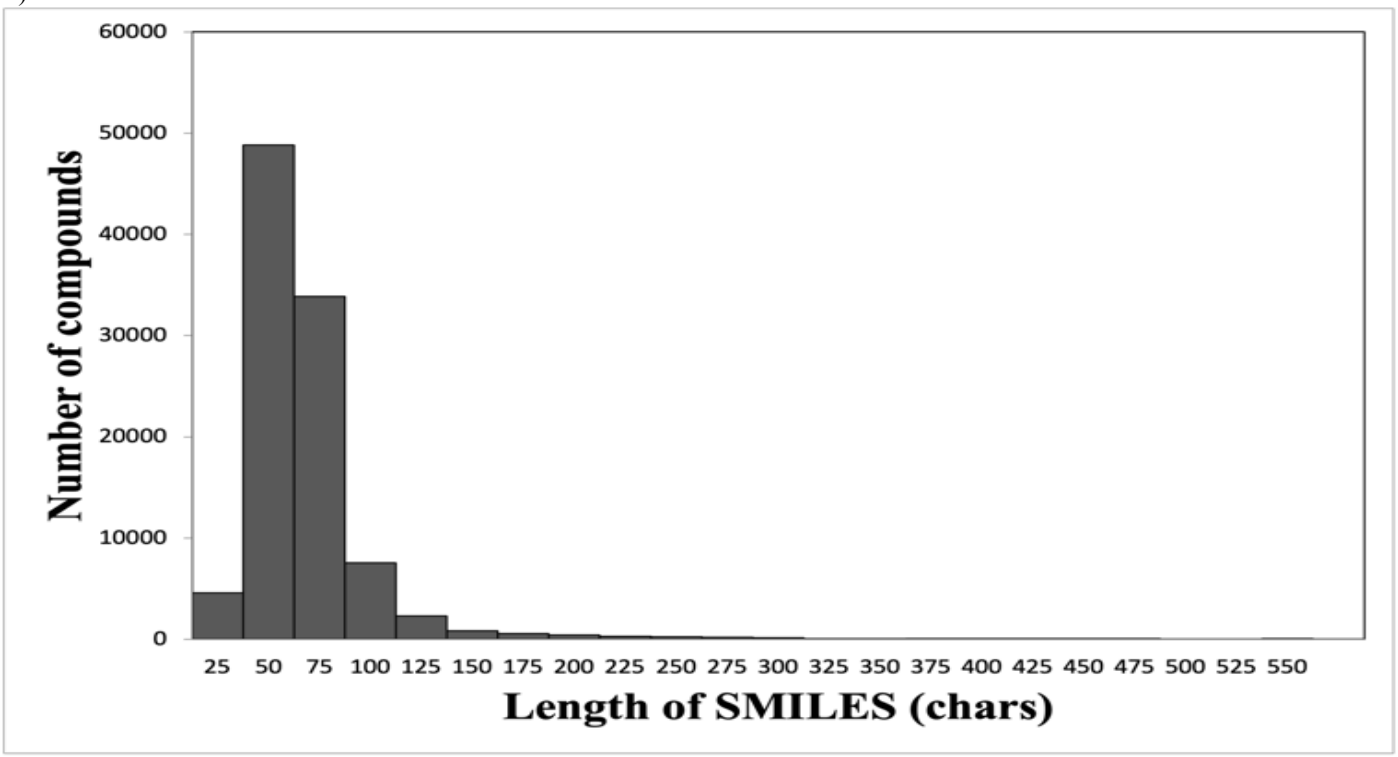


Figure 2.1.1.: Summary of the Binding Affinity Dataset. (A) Range of pKd values. (B) Range of character lengths of the protein sequences. (C) Range of character lengths of the SMILES strings.

AFBAP was trained on a randomly chosen subset of 100,000 samples of the Binding Affinity Dataset [24], which originally contained 1.9M distinct pairs of ligands in the form of SMILES strings, proteins in the form of amino acid sequences, and their corresponding binding affinities in the normalized negative base 10 form, which were determined experimentally. The seed 42 was used to select the data points in Python using NumPy. SMILES (Simplified Molecular Input Line Entry System) allows for the representation of ligands as 1D strings, suitable for high-speed machine processing [25]. A reduced set was used due to the existence of computational constraints.

The model's performance was verified using two benchmarking datasets: the Test2016_290 dataset from Jin et al. [14], and the CSAR-HiQ-36 dataset [27]. The Test2016_290 dataset was used for general benchmarking purposes, and it was confirmed that all data points were unique and that there were no similar data points between the datasets used for training and validating the model. Similarity was calculated using the Smith-Waterman similarity [26] test, and it was found that no two pairs of proteins and ligands were closer than 60% for 99% of the data points in the dataset. The CSAR-HiQ-36 dataset [27] measured the generalization capabilities of the model and consisted of 36 protein-ligand pairs.

2.2 Data Pre-processing

In order to extract higher-dimensionality data from the one-dimensional string inputs, the ProtBert and ChemBERTa models were utilized [23, 19]. They accepted the canonical SMILES strings and amino acid sequences as input, and produced averaged feature embeddings as output, with 1024 dimensions for the protein input, and 768 dimensions for the ligand, which when combined resulted in a concatenated feature vector with 1792 dimensions, which was used for training the prediction module. This process was carried out on an M2 Macbook Air with 16GB of RAM.

This process was prefaced by randomly splitting the dataset into training and validation sets, with 90,000 data points for the training set (90%) and 10,000 data points for the validation set (10%). This was done randomly using the NumPy library and the seed 103, which shuffled and then split the dataset.

2.3 Input Representation

As a sequence-based machine learning model, AFBAP takes in one-dimensional string inputs in the form of SMILES strings for ligands and amino acid strings for proteins, and then subsequently predicts the binding affinity. Unlike prior models like CAPLA that utilized transformers, AFBAP does not utilize protein packet data [14].

***Protein Input Processing*:** As mentioned previously, AFBAP accepts proteins as one-dimensional amino acid string inputs. The protein processing pipeline is a four-step process: first, we separate the inputted characters using spaces. Then, we replace non-standard amino acids (B, U, Z, O, J) with the character 'X'. We then tokenize the edited sequence in accordance to the methodology utilized by Elnaggar et al. in their paper regarding ProtBERT [28, 29]. Tokenization is the process of transforming a text sequence into a model-recognizable format. This is done by building a model-specific vocabulary, where sub-sequences of the text are mapped to unique integer IDs.

By segmenting the text into smaller parts and assigning each a unique identifier, the model can efficiently interpret and process the input. The output of these steps are a 3200 character long token list that is padded, along with an attention mask of the same length. While ProtBERT [19] can handle sequences up to 40,000 in length, computational limitations when dealing with larger inputs necessitate adjusting the sequence length to 3200 tokens through padding or truncation. As only 0.3% of the protein-ligand data points exceed this length, this is not expected to noticeably impact model performance.

***Molecule Input Processing*:** For chemical compounds, AFBAP uses canonical SMILES strings as input. This choice capitalizes on the ability of transformers to deal with language-like input, as SMILES strings are similar to the normal English language, which integrates well with existing frameworks. One-dimensional inputs leverage the ability of transformers to process tokens and recognize complex patterns to a great extent, achieving performance comparable to, and sometimes even better than three-dimensional ligand representations.

The SMILES strings were tokenized using Schwaller et al's SmilesTokenizer[30], using both padding and truncation to achieve a uniform token sequence length of 278. Similar to the character length restriction on ProtBert, although ChemBERTa can accept strings up to 512 characters in length, our longest sequence was only 278 characters long, thus the padding. After pre-processing the SMILES inputs using the SmilesTokenizer, the output consisted of the tokenized strings and a custom attention mask, which was the input for the ChemBerta model.

2.4 Model Architecture

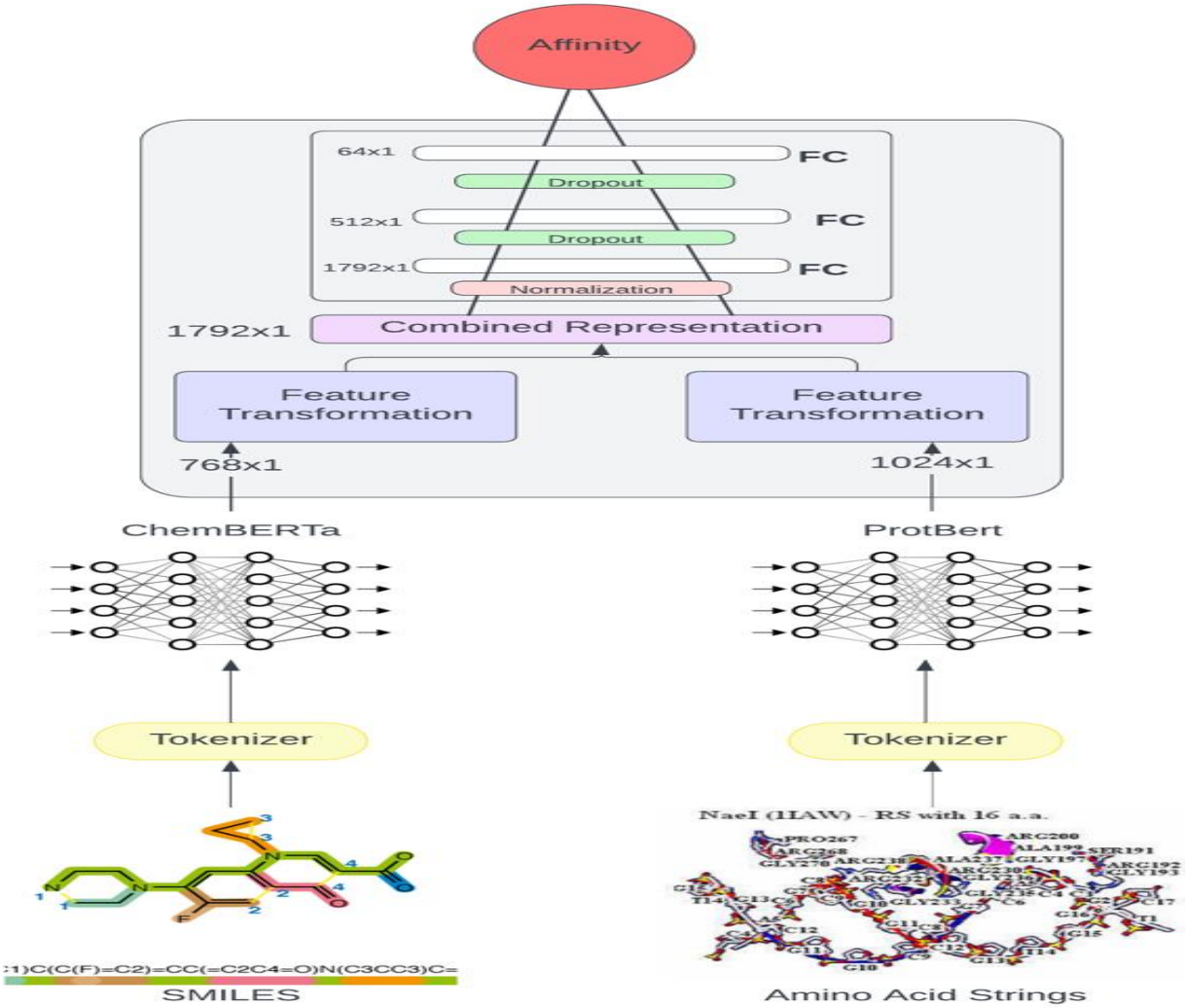


Figure 2.4.1: Diagram representation of AFBAP. The SMILES and amino acid strings are initially tokenized, then passed into ChemBERTa and ProtBert, the output of which is concatenated and passed through a task-adaptive feature transformation followed by fully connected layers with a normalization and two dropout layers.

AFBAP's model architecture consists of to primary parts: a pair of BERT-like transformer encoders that perform feature extraction and produce feature embeddings, along with a novel Adaptive Fully-Connected Layer architecture that functions as a prediction module.

2.4.1. *Feature Extraction*

Feature extraction can be implemented in one of two ways – in a combined manner, where both the protein and ligand have their features extracted as a part of the same input, or separately, where individual features are extracted, and then subsequently concatenated and used. Although studies like CAPLA utilize the first method to great effect in tandem with a cross-attention mechanism [14], it is much more straightforward for us to utilize the second method since we can then make use of the improved knowledge and generalizability of our pre-trained transformer encoders, and then concatenate the two feature embeddings before affinity prediction.

ProtBert is utilized for protein feature extraction, and the reason this transformer was chosen was due to its superior performance in generating embeddings, high precision in predicting secondary structures, determining subcellular locations, and addressing various other tasks [21, 23]. The model accepts amino acid sequences with an attention mask and token list 3200 characters long (as mentioned in Section 2.2) as input, and generates an 1024-dimensional output vector that contains structural feature data.

For extraction of ligand features, AFBAP utilizes the ChemBERTa model, which accepts SMILES strings as input. ChemBERTa was chosen for its ability to generate powerful molecular feature extractions that aid in predicting downstream molecular properties [19]. It generates a vector with 768 dimensions as output, which contains information about the molecular structure and important attributes of the inputted ligand.

The outputs of both ProtBert and ChemBERTa are concatenated to form a 1792-dimensional output vector, which is the input for the subsequent Adaptive Fully-Connected Layer architecture. After passing this concatenated input to the prediction module, it generates a predicted binding affinity value.

2.4.2. *Prediction Module*

AFBAP's prediction module consists of two parts: a Convolutional Neural Network (CNN), and a task-adaptive feature transformation that prefaces it. The task-adaptive feature transformations reduces the dimensionality of the input and focuses on important features, while the CNN's Fully Connected (FC) layers are trained to predict the final binding affinity. As mentioned before, the input to the Prediction Module is the concatenated 1024-dimensional feature vector produced by ProtBert and ChemBerta, which is subsequently split into its original components (a 1024-dimensional vector for proteins and a 768-dimensional vector for ligands), and fed into the prediction module for subsequent processing and training.

Once split, both the feature vectors undergo a task-adaptive feature transformation for one-shot learning, as proposed by Ziko et al. [31]. This transformation aims to reduce focus onto fewer relevant dimensions and reduces overfitting by tamping down on the influence of non-essential feature dimensions.

The final produced features are again concatenated, normalized using PyTorch, and passed into the CNN, which consists of three Fully Connected (FC) layers. Normalization, a pre-processing technique, scales or transforms data to ensure each feature contributes equally by mapping the values of each feature set onto a uniform interval [32]. The FC layers contain 1792, 512 and 64 nodes, in order, each with a dropout rate of 0.2. In between each layer, we employ *dropout layers,* which randomly set the activation of certain neurons to 0 to prevent overfitting. We use ReLU (Rectified Linear Unit) as the activation function for the FC layers [33]. The equation for ReLU is $g(x) = \max(0, x)$. The final output is give in the form of a dissociation constant (K_d), for which higher values represent stronger binding strength, and lower values represent weaker binding strengths. The goal of the model is to get the predicted value as close to the real binding affinity value as possible. The loss function employed is mean squared error (MSE), which is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(P_i - Y_i)^2$$

where P is the predicted value, Y is the actual value, and n is the number of samples.

2.5 Training

The model was trained for 50 epochs, with a mini-batch size of 256. Adam was the optimization algorithm used [34], with a default learning rate of 0.001. PyTorch's Embedding layer was used to represent the high-dimensional inputs. An M2 Macbook Air with 16 GB RAM was used as the training platform. Compute details can be found in Appendix A.

2.6 Evaluation Metrics

AFBAP's accuracy was measured using three metrics: Pearson correlation coefficient (R), root mean square error (RMSE), and mean absolute error (MAE). Predicted values were compared against real values, which meant that model output had to be denormalized before processing and analysis.

Pearson's Correlation Coefficient (R): This measures the linear correlation between two datasets. It is the ratio between the covariance of two variables and the product of their standard deviations.

$$r = \left(\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \right)$$

Here, n is the sample size, x_i, y_i are the individual observed and predicted sample points indexed with i, and $\bar{x}$ and $\bar{y}$ represent the sample means.

Root Mean Square Error (RMSE): If $x_i \dots x_n$ represent the experimental values, $y_i \dots y_n$ represent the computed values, and n represents the number of data points, then the RMSE, which measures how far predictions fall from measured true values, is represented by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

Mean Absolute Error (MAE): MAE is a is a measure of total errors between paired observations expressing the same phenomenon.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|$$

Here, y_i is the prediction, x_i is the true value, and n is the number of number observatinons.

AFBAP's performance was evaluated against 10 other models: OnionNet [12], DeepDTAF [13], GLI [38], CAPLA, Pafnucy [11], IMCP-SF [37], DeepDTA [35], FAST [36], CAPLA-Pred [14] and Blanchard et al's model to predict ligands to develop vaccines for SaRS-COV-2. The quantified comparisons are made using the prior defined benchmark datasets, providing an accurate assessment of AFBAP's accuracy in relation to other models in the field.

3.RESULTS

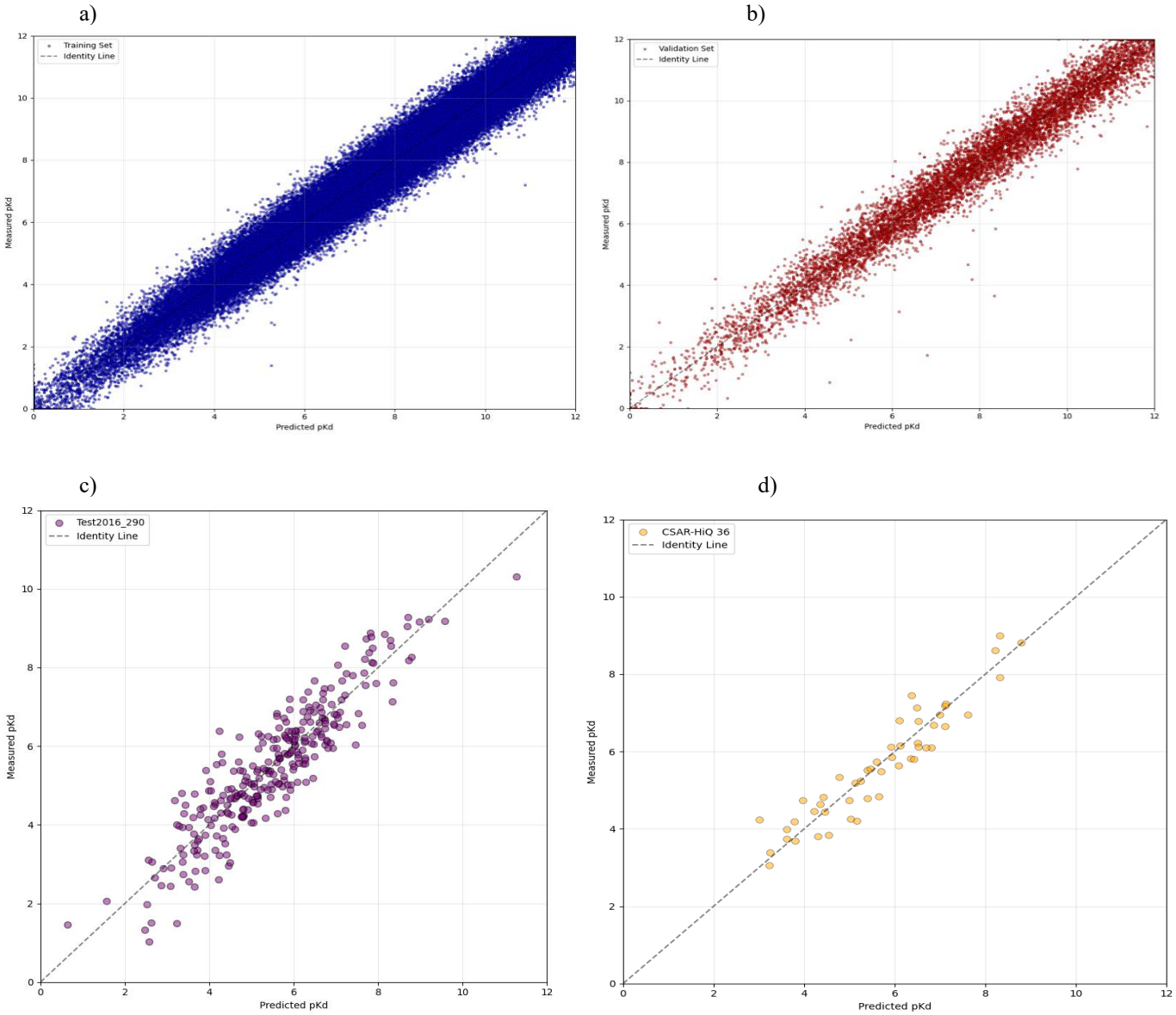


Figure 3.1: AFBAP Benchmark Performance on following datasets: (a) Training Set, (b) Validation Set, (c) Test2016_290, and (d) CSAR-HiQ_36.

This section presents and examines AFBAP's performance metrics. We begin by presenting and analysis training, validation and test set results. AFBAP is then subsequently benchmarked against Test2016_290 and CSAR-HiQ_36, which showcases AFBAP's performance, reduced need for computational resources, and capacity for generalization.

3.1 Performance of AFBAP

Table 3.1.1 puts forth AFBAP's benchmark metrics on the training and validation set, utilising the Pearson correlation coefficient (R), root mean square error (RMSE), and mean absolute error (MAE). AFBAP achieves state-of-the-art performance on the training dataset, with an R value of 0.881, and negligible error with an RMSE of 0.776 and Mean Absolute Error of 0.771. A concern might be potential overfitting of the model to the training set, as seen by the drop in the R value from 0.881 in the training set to 0.743 in the validation set. Nevertheless, when compared to other models like CAPLA with an RMSE of 1.338 and MAE of 0.134 [14], validation RMSE and MAE is notably lower. This offers very compelling evidence as to the fact that AFBAP's level of overfitting is at most equal to current state-of-the-art models, and in all likeliness even lower.

Table 1: AFBAP Model's Baseline Benchmarks

Data Set	R (↑)	MSE (↓)	RMSE (↓)	MAE (↓)
Training Data	0.881	0.587	0.776	0.771
Validation Data	0.743	1.462	1.221	0.945

Note: ↑ indicates that the greater the value, the better, and ↓ indicates that the greater the value, the worse.

3.2 Comparative Analysis

AFBAP's capabilities were benchmarked using the CSAR-HiQ_36 [28] and Test2016_290 [14] datasets in order to accurately evaluate relative performance to other state-of-the-art models.

Test2016_290 Dataset: This dataset was curated by Jin. et al [14], and was the same model used to benchmark CAPLA. AFBAP's superior performance is evident from table 3.2.1, as it outperforms a variety of current architectures across a number of benchmarking metrics. The only new metric introduced is Time, which refers to the training duration of an epoch on the dataset, which contains 11,906 samples processed in our experimental environment. It should be highlighted that Blanchard et al.'s affinity prediction model [22] surpassed AFBAP across all metrics for this benchmark. Further investigation disclosed that a number of data points in the Test2016_290 dataset, 177 out of the 290 to be exact, were part of the datasets used to train and validate the model used by Blanchard et. al, which raises concerns about overfitting and led to their exclusion from this comparison.

Table 3.2.1: AFBAP vs. Other Models [11, 10, 8, 9, 33, 34] on the Test2016_290 Benchmark [14]

Method	R (↑)	RMSE (↓)	MAE (↓)	Time (s) (↓)
AFBAP	**0.851**	**1.185**	**0.921**	**26**
CAPLA	0.843	1.200	0.966	27
DeepDTA	0.749	1.443	1.148	47
DeepDTAF	0.789	1.355	1.073	26
Pafnucy	0.775	1.418	1.129	2894.3
FAST	0.810	1.308	1.019	82.1
IMCP-SF	0.791	1.452	1.155	635.9

CSAR-HiQ_36 Benchmark: When benchmarked on the CSAR-HiQ_36 dataset [29], we see AFBAP demonstrating excellent performance across all metrics, outperforming other models such as Pafnucy, DeepDTAF and IMCP-SF [11, 13, 37], as corroborated by table **3.2.1**. However, Blanchard et. al's affinity_pred model exhibits a higher R value of R = 0.774 as compared to AFBAP's R = 0.735. Despite this, AFBAP is more accurate, with RMSE values improved by 9.29% and MAE values improved by 2.12%.

Table 3.2.1: AFBAP vs. Other Models [9, 11, 35, 24, 8, 34] on the CSAR-HiQ_36 Benchmark [29]

Method	R (↑)	RMSE (↓)	MAE (↓)
AFBAP	0.735	**1.346**	**1.151**
Pafnucy	0.566	1.658	1.291
CAPLA	0.704	1.454	1.160
IGN	0.528	1.795	1.431
affinity_pred	**0.774**	1.484	1.176
DeepDTAF	0.543	2.765	2.318
IMCP-SF	0.631	1.560	1.205

4. Conclusion

This study introduces the Attention For Binding Affinity Prediction (AFBAP) model, a novel machine learning approach designed to predict protein-ligand binding affinities with high accuracy and computational efficiency. The results demonstrate AFBAP's superior performance across multiple benchmarks, positioning it at the forefront of current methodologies in the field.

AFBAP's success can be attributed to its innovative architecture, which uses pre-trained encoders like ProtBert and ChemBerta to leverage their transfer learning effects along with an Adaptive Fully-Connected Layer. This design enables efficient prediction from one-dimensional sequence data, representing a significant advancement in the field of computational drug discovery.

The implications of AFBAP extend beyond its immediate predictive capabilities. In the context of drug discovery and design, accurate binding affinity predictions are crucial for identifying promising drug candidates efficiently. By reducing the computational resources required for training, AFBAP democratizes access to advanced predictive tools, potentially accelerating the pace of drug discovery across a broader range of research institutions. However, it is important to acknowledge the limitations of the current implementation. The availability of more computation resources would allow the use of a more extensive dataset, enhancing performance. These limitations, while not detracting from AFBAP's current performance, indicate potential areas for improvement in subsequent iterations of the model. Future research directions for AFBAP are multifaceted. Binding site data would be included in training the model, which could potentially increase the model's predictive accuracy by providing additional contextual information. Another area of exploration could be the development of better pre-trained protein transformers or variations of different feature transformations to determine which one focuses the best. Furthermore, testing AFBAP using more datasets and in real world applications would be valuable in assessing its robustness and generalizability. This expanded validation could include datasets representing a broader spectrum of protein-ligand interactions, potentially uncovering new insights into molecular binding mechanisms. The development of AFBAP represents a significant step forward in the application of machine learning to molecular interactions. As the field continues to advance, models like AFBAP are likely to play an increasingly important role in bridging the gap between computational predictions and experimental validation. In conclusion, AFBAP exemplifies the transformative potential of innovative machine learning approaches in biotechnology. By combining advanced pre-trained models with efficient architecture, it offers a powerful new tool for researchers working on developing new drugs. With the growing relevance of artificial intelligence in molecular biology, models like AFBAP are poised to exponentially increase the rate at which discoveries are made, potentially leading to more rapid identification of novel therapeutic agents. The success of AFBAP underscores the ongoing importance of computational approaches in addressing complex biological questions and highlights the potential for further advancements in this rapidly evolving field.

References

[1] Clark, L. J., Walton, J. W., & Jones, G. (2021). MicroED for the study of protein–ligand interactions and the potential for drug discovery. Nature Reviews Chemistry, 5(12), 853-858.
[2] Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., ... & Collins, J. J. (2020). A deep learning approach to antibiotic discovery. Cell, 180(4), 688-702.
[3] Gapsys, V., Pérez-Benito, L., Aldeghi, M., Seeliger, D., Van Vlijmen, H., Tresadern, G., & de Groot, B. L. (2020). Large scale relative protein ligand binding affinities using non-equilibrium alchemy. Chemical Science, 11(4), 1140-1152.
[4] Wang, H., Liu, X., Xu, Y., Peng, J., Zeng, J., & Shen, H. B. (2022). DLSSAffinity: protein–ligand binding affinity prediction via a deep learning model. Physical Chemistry Chemical Physics, 24(16), 10124-10133.
[5] Trott, O., & Olson, A. J. (2010). AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of Computational Chemistry, 31(2), 455-461.
[6] Allen, W. J., Balius, T. E., Mukherjee, S., Brozell, S. R., Moustakas, D. T., Lang, P. T., ... & Rizzo, R. C. (2015). DOCK 6: Impact of new features and current docking performance. Journal of Computational Chemistry, 36(15), 1132-1156.
[7] Ruiz-Carmona, S., Alvarez-Garcia, D., Foloppe, N., Garmendia-Doval, A. B., Juhos, S., Schmidtke, P., ... & Morley, S. D. (2014). rDock: A fast, versatile and open source program for docking ligands to proteins and nucleic acids. PLoS Computational Biology, 10(4), e1003571.
[8] Pagadala, N. S., Syed, K., & Tuszynski, J. (2017). Software for molecular docking: A review. Biophysical Reviews, 9(2), 91-102.

[9] Rube, H. T., Rastogi, C., Feng, S., Kribelbauer, J. F., Li, A., Becerra, B., ... & Bussemaker, H. J. (2022). Prediction of protein–ligand binding affinity from sequencing data with interpretable machine learning. Nature Biotechnology, 40(10), 1520-1527.
[10] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
[11] Stepniewska-Dziubinska, M. M., Zielenkiewicz, P., & Siedlecki, P. (2018). Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. Bioinformatics, 34(21), 3666-3674.
[12] Zheng, L., Fan, J., & Mu, Y. (2019). OnionNet: A multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. ACS Omega, 4(14), 15956-15965.
[13] Wang, K., Zhou, R., Li, Y., & Li, M. (2021). DeepDTAF: A deep learning method to predict protein–ligand binding affinity. Briefings in Bioinformatics, 22(5), bbab072.
[14] Jin, Z., Wu, T., Chen, T., Pan, D., Wang, X., Xie, J., ... & Lyu, Q. (2023). CAPLA: Improved prediction of protein–ligand binding affinity by a deep learning approach based on a cross-attention mechanism. Bioinformatics, 39(2), btad049.
[15] Monteiro, N. R. C., Oliveira, J. L., & Arrais, J. P. (2022). DTITR: End-to-end drug–target binding affinity prediction with transformers. Computers in Biology and Medicine, 147, 105772.
[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.
[17] Zhang, S., Fan, R., Liu, Y., Chen, S., Liu, Q., & Zeng, W. (2023). Applications of transformer-based language models in bioinformatics: A survey. Bioinformatics Advances, 3(1), vbad001.
[18] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., ... & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 379(6637), 1123-1130.
[19] Chithrananda, S., Grand, G., & Ramsundar, B. (2020). ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction. arXiv preprint arXiv:2010.09885.
[20] Kimber, T. B., Chen, Y., & Volkamer, A. (2021). Deep learning in virtual screening: Recent applications and developments. International Journal of Molecular Sciences, 22(9), 4435.
[21] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).
[22] Blanchard, A. E., Gounley, J., Bhowmik, D., Shekar, M. C., Lyngaas, I., Gao, S., ... & Glaser, J. (2022). Language models for the prediction of SARS-CoV-2 inhibitors. The International Journal of High Performance Computing Applications, 36(5-6), 587-602.
[23] Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., & Linial, M. (2022). ProteinBERT: A universal deep-learning model of protein sequence and function. Bioinformatics, 38(8), 2102-2110.
[24] Glaser, J. (2021). Binding affinity dataset. https://huggingface.co/datasets/jglaser/binding_affinity
[25] Weininger, D. (1988). SMILES, a chemical language and information system. Journal of Chemical Information and Computer Sciences, 28(1), 31-36.
[26] Pearson, W. R. (1991). Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics, 11(3), 635-650.
[27] Dunbar Jr, J. B., Smith, R. D., Yang, C. Y., Ung, P. M. U., Lexa, K. W., Khazanov, N. A., ... & Carlson, H. A. (2011). CSAR benchmark exercise of 2010: Selection of the protein–ligand complexes. Journal of Chemical Information and Modeling, 51(9), 2036-2046.
[28] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., ... & Rost, B. (2020). ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. arXiv preprint arXiv:2007.06225.
[29] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., ... & Rost, B. (n.d.). prot_bert.
[30] Schwaller, P., Probst, D., Vaucher, A. C., Nair, V. H., Kreutter, D., Laino, T., & Reymond, J. L. (2020). Mapping the space of chemical reactions using attention-based neural networks. ChemRxiv.
[31] Ziko, I. M., Lecue, F., & Ayed, I. B. (2023). Task adaptive feature transformation for one-shot learning. arXiv preprint arXiv:2304.06832.
[32] Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97, 105524.
[33] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 807-814).
[34] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference for Learning Representations, San Diego.
[35] Öztürk, H., Özgür, A., & Ozkirimli, E. (2018). DeepDTA: Deep drug–target binding affinity prediction. Bioinformatics, 34(17), i821-i829.
[36] Jones, D., Kim, H., Zhang, X., Zemla, A., Stevenson, G., Bennett, W. F. D., ... & Allen, J. E. (2021). Improved protein–ligand binding affinity prediction with structure-based deep fusion inference. Journal of Chemical Information and Modeling, 61(4), 1583-1592.
[37] Wang, D. D., & Chan, M. T. (2022). Protein-ligand binding affinity prediction based on profiles of intermolecular contacts. Computational and Structural Biotechnology Journal, 20, 1088-1096.
[38] Zhang, Y., Zhou, G., Wei, Z., & Xu, H. (2022). Predicting protein ligand binding affinity via joint global-local interaction modeling. arXiv preprint.
[39] Jiang, D., Hsieh, C. Y., Wu, Z., Kang, Y., Wang, J., Wang, E., ... & Wu, J. (2021). InteractionGraphNet: A novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. Journal of Medicinal Chemistry, 64(24), 18209-18232.

Appendix A – Compute

One computer was used to train the AFBAP model. Both feature extraction and model training were done on the same machine.

Table 4: Specifications of the computer used to train and perform feature extraction for the AFBAP model

Specification	Detail
CPU	8-core Macbook M2 CPU
GPU	8-core Macbook M2 GPU
RAM	16 GB
Operating System	MacOS Sonoma 14.1.1

Appendix B – Training Details

Training was done using Python (PyTorch), using the model.train() function. Training code can be found on AFBAP's Github Repository, linked in Appendix C.

Table 4: List of Training Hyperparameters

Specification	Detail
Loss Function	MSE
Optimizer	Adam (Decay $1e^{-5}$)
Learning Rate	0.001
Batch Size	256
Max Training Rounds	50

Appendix C – Supplementary Materials

C.1 Source Code

The source code used for data pre-processing, feature extraction and model training can be found at AFBAP's Github Repository. The link can be found below:

https://github.com/HariharPrasadd/AFBAP/

C.2 Datasets

The training dataset for AFBAP was the Binding Affinity Dataset by Glaser et. al [24]. The link can be found below:

https://huggingface.co/datasets/jglaser/binding_affinity