Machine Learning and Deep Learning models for prediction of Protein-Ligand Binding Affinity

by
Parneet Kaur
MT22193

Under the Supervision of Dr. N. Arul Murugan
Indraprastha Institute of Information Technology, Delhi
May, 2024

Machine Learning and Deep Learning models for prediction of Protein-Ligand Binding Affinity

By
Parneet Kaur
(MT22193)

Submitted
in partial fulfillment of the requirements for the degree of Master of Technology

To
Indraprastha Institute of Information Technology Delhi Month, 2024

CERTIFICATE

This is to certify that the thesis titled *"Machine Learning and Deep Learning models for prediction of protein-ligand binding affinity"* being submitted by Parneet Kaur (MT22193) to the Indraprastha Institute of Information Technology, Delhi, for the award of the Master of Technology, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.
The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

May, 2024, Dr. N. Arul Murugan

Department of Computational Biology

Indraprastha Institute of Information Technology, Delhi

New Delhi 110 020

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor, Dr. N Arul Murugan, for his valuable mentorship and constant support throughout the work. The completion of this thesis would not have been possible without his expert knowledge and encouragement. I would like to thank all the faculty and staff at IIIT, Delhi, for their consistent assistance throughout the entire process. I am thankful to my fellow batchmates for their encouragement and support. Finally, I would like to thank my family, who have been my constant source of strength and inspiration, and without them, this academic journey would not have been possible.

ABSTRACT

In recent years, there has been significant interest in using Machine Learning and Deep Learning to predict protein-ligand binding affinity. This is due to the rapid growth of the computational approaches that have evolved in drug discovery. The binding affinity prediction is useful in the virtual screening and drug screening optimization step of drug discovery.. The ML and DL-based approaches have shown notable improvements compared to the conventional approaches. The conventional approaches are time-consuming, complex, and challenging. However, the introduction of computational approaches has expedited the drug discovery timeline. In this study, we aim to develop Machine Learning models and benchmark some of the Deep Learning models to predict the protein-ligand binding affinity. We have used the refined set of the PDBbind database(version 2020) to fetch the protein-ligand structural data and binding affinity data. We have used the dataset mentioned above for the machine learning models and featurized the protein-ligand complexes using tools such as RDkit/Mordred and Pfeature, followed by feature selection. Models such as SVM, Random Forest, Multiple Linear Regression, etc, have been used to predict the binding affinity of PL complexes. From all the ML models we tested, it was observed that Random Forest performed better with an R-squared value of 0.6. Further, we benchmarked the CNN-based Deep learning models such as Pafnucy and OnionNet-2 using the refined set of PDBbind as the benchmarking test dataset. It was observed that the OnionNet-2 model showed better predictive performance at an R-squared value of 0.85 than that of the Pafnucy model at an R-squared value of 0.46. We have discussed this relative performance in our study. Hence, it was observed that out of all the approaches we used, the PDBbind refined dataset showed the maximum R-squared value when it was benchmarked using the OnionNet-2 model. We have also discussed the reasons for the variation and the future scope of the study.

Keywords: *binding affinity, protein-ligand complex, PDBbind, machine learning, deep learning*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADMET: Absorption, Distribution, Metabolism, Excretion, Toxicity
ANN: Artificial Neural Network
ATP: Adenosine Triphosphate
BA: Binding affinity
CADD: Computer-Aided Drug Design
CNN: Convolutional Neural Network
DNA: Deoxyribonucleic Acid
DL: Deep Learning
DNN: Deep Neural Network
GNN: Graph Neural Network
HOMO: Highest occupied molecular orbital
IC50: Half-maximal Inhibitory Concentration
Ka: Association Constant
Kd: Dissociation Constant
Ki: Inhibition Constant
LBDD: Ligand-based Drug Design
LUMO: Lowest unoccupied molecular orbital
MD: Molecular Docking
MDS: Molecular Dynamics Simulations
ML: Machine Learning
MLR: Multiple Linear Regression
NN: Neural Network
PDB: Protein Data Bank
PL: Protein-ligand
QSAR: Quantitative Structure-Activity Relationship
RF: Random Forest
RNA: Ribonucleic Acid
SBDD: Structure-based Drug Design
SF: Scoring Function
SVM: Support Vecoor Machine
SVR: Support Vector Regressor

INTRODUCTION

1. Ligands and Proteins

Proteins are complex structures formed by the polymerization of amino acids. The amino acids are connected by polypeptide bonds. Proteins are vital to a number of biological processes in all living organisms. Proteins have a variety of functions that are responsible for maintaining the normal working and maintenance of the organisms. Some of the functions of the proteins include the transportation of cargo molecules, catalysis, membrane transfer, information exchange, etc[1]. However, proteins are not capable of acting independently to carry out the activities as mentioned above and hence, they need to associate with other proteins or molecules such as RNA, DNA, or small organic/inorganic molecules, to function correctly[2]. For example, protein-based enzymes such as human kinases must be bound to ATP (Adenosine Triphosphate) to carry out various signaling and metabolic functions and processes[3]. Mutations in the residues of kinase proteins may cause abnormalities in the activity of kinases, resulting in diseases such as cancer[4], Parkinson's[5], etc. Therefore, it is crucial to develop small molecules that can target the kinase proteins so as to assist in the regulation of its activity and treat diseases. These small molecules are referred to as drugs or ligands in biochemistry. The specific residues in the protein which interact with the ligands are called binding pockets[6]. Binding pockets are concave in shape and facilitate the molecular recognition between proteins and ligands. The small molecule drugs/ligands exert their effect when these are tightly bound to the protein pocket and hence exhibit a lock and key type of modular structure. This structure formed as a result of interaction between the protein and ligand molecule is called a *protein-ligand complex*[7]. Pockets are an essential area of research regarding drug discovery and docking studies. The binding of ligands to the pockets is defined by a number of interactions, such as hydrogen bonding, hydrophobic interactions, and electrostatic interactions, to name a few. The binding strength of the protein pocket and ligands can be represented quantitatively by the term *binding affinity*[8]. Binding affinity is a critical aspect of the drug discovery pipeline, as one of the most crucial steps is to identify the ligands/drug molecules that bind to the target protein with a high binding affinity. Hence, BA is an integral part of the drug screening process and virtual screening.

2. Protein-ligand complex

Ideally, the drug target, aka the target protein, must be associated with a specific disease and have a concave, pocket-like surface property where the drug can bind. The structure formed due to the binding of the drug/ligand and target is called a protein-ligand complex formed as a result of molecular recognition between the ligand and the protein[6]. The nature of the interaction is specific and hence plays a vital role in various biological functions.

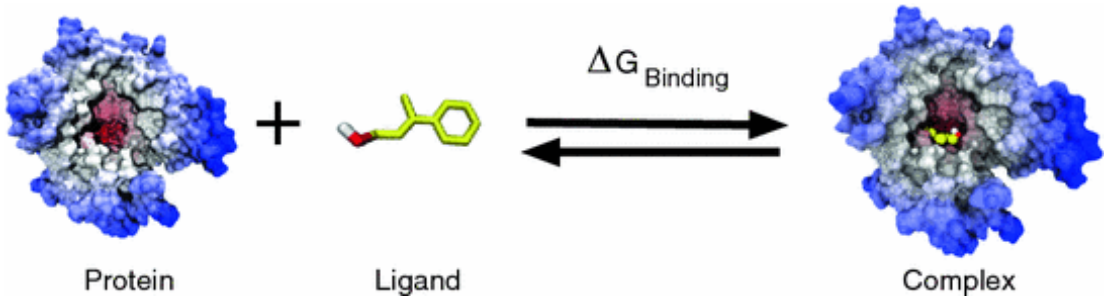

Figure 1: Diagrammatic representation of protein-ligand complex using docking studies. Source: [9]

3. Binding Affinity of protein-ligand complex

Binding affinity can be defined as the quantitative value used to measure the strength of binding between two molecules that bind reversibly. Every PL complex exhibits binding affinity, which can be used in the screening stages of the drug discovery process eventually leading to the identification of the lead compounds. The affinity value can be determined with the help of a term known as dissociation constant or Kd[10]. It can also be represented as the inverse of the association constant term given by (1/Ka). The Kd and (1/Ka) describe the nature of protein-ligand binding in terms of complexed or uncomplexed.

$$[\mathrm{A}] + [\mathrm{B}] \underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} [\mathrm{AB}],$$

$$K_{\mathrm{d}} = \frac{[\mathrm{A}][\mathrm{B}]}{\mathrm{AB}} = \frac{k_{\mathrm{off}}}{k_{\mathrm{on}}}.$$

Figure 2: Representation of Kd (Dissociation constant) showing an example of receptor(protein) and ligand(drug).

4. Drug Discovery

The principal goal of the drug discovery process is identifying specific molecules known as drugs that selectively interact with the biological targets or receptor proteins to modulate a disease. The general timeline of a drug discovery process is around 8-10 years and can cost up to Billions of Dollars[11]. Hence, the process is expensive in terms of monetary value and time. Only a select few drugs come into the market every year after going through the lengthy process of drug discovery. The vastness of the chemical space makes this fact appear very miniscule. Other than affinity, the drug candidate is also required to exhibit specific properties such as specificity, affinity, ADMET(Absorption, Distribution, Metabolism, Excretion, and Toxicity), etc[12]. Since the pipeline of drug discovery is very time-intensive, traditional and experimental approaches are now being gradually replaced by computational methods. These methods are quite helpful in reducing both the cost and the time consumed in the drug prediction process. The process of computational drug design is described by an umbrella term known as Computer-aided drug design/discovery[13]. The methods that come under CADD can be broadly classified into two categories i.e., Structure-based and ligand-based methods.

5. SBDD and LBDD

SBDD and LBDD are the abbreviations used for the two categories of CADD, where SBDD stands for Structure-based drug design, and LBDD stands for Ligand-based drug design.

- SBDD: Structure-based methods rely on the target protein structure knowledge to optimize the candidate molecules to be tested[14]. If the three-dimensional structure of the protein receptor target of a specific disease is known beforehand, in that case SBDD methods are employed. An example could be a soluble protein which can be crystallized. Some of the methods of SBDD are molecular docking, de novo design, molecular dynamics simulation, and structure-based virtual screening.

- LBDD: On the other hand, if the 3D structure of the protein target is not known, especially in the case of membrane proteins, then the molecule of interest becomes the ligand[15]. In such a case, the knowledge of molecules, whether active or inactive, is exploited by employing chemical search mechanisms and QSAR(Quantitative Structure-Activity Relationships) models[16]. Some of the

methods of LBDD are QSAR, pharmacophore modeling, similarity searching, machine learning, and ligand-based virtual screening.

6. CADD in the Drug Discovery Pipeline

The hit rate of a new drug can be increased utilizing CADD-based approaches due to its intensive targeted nature[17]. In a drug discovery process, CADD can be used to cater to the following purposes:

- From an extensive chemical library, filter the compounds into smaller sets based on their activity that can be further filtered.
- Lead compound optimization either binding affinity or DMPK(drug metabolism and pharmacokinetics) properties such as ADMET (absorption, distribution, metabolism, excretion and toxicity)[18].
- Designing novel compounds either by growing chain method or by putting together fragments.

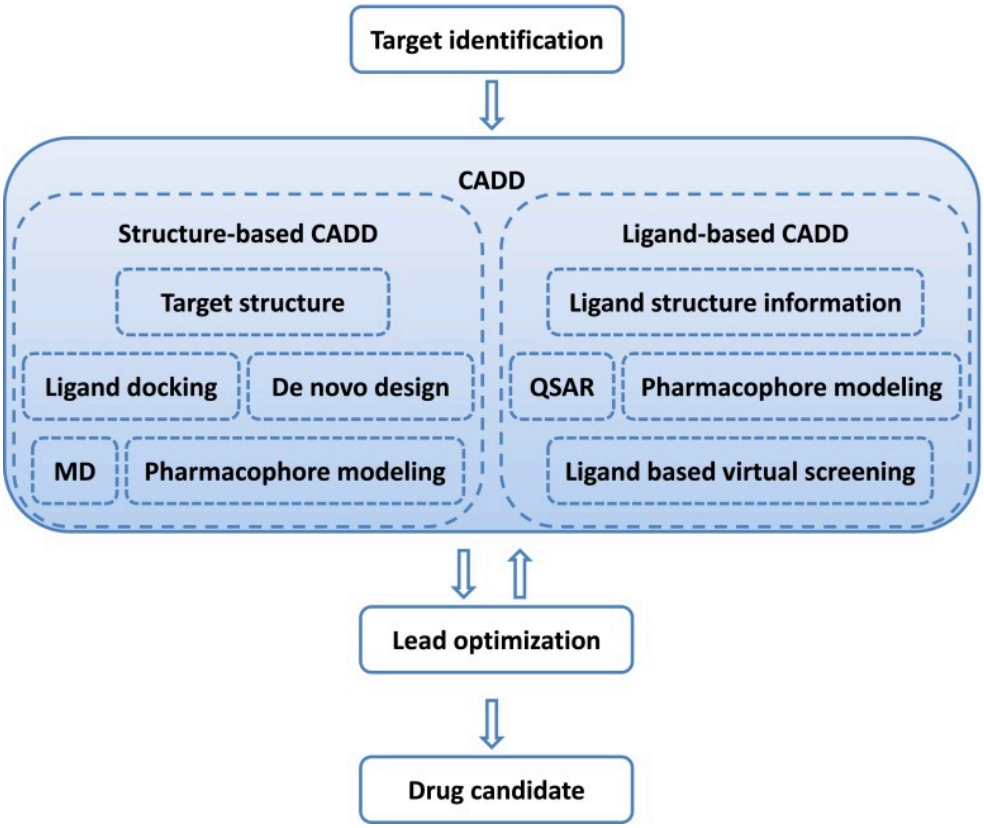

Figure 3: Workflow of CADD in the Drug discovery process[19]

7. Scope of the current study: Predicting the protein-ligand interactions

Traditional CADD methods such as molecular docking, molecular simulations, and structure-based virtual screening (SBVS) are undertaken to search for the molecules that bind to their respective protein receptors[20]. The process of evaluation of a particular ligand against a protein is done by scoring. It is done by taking into account the, favourable intermolecular interactions between the ligand and the protein. Hence, the mathematical functions which are utilized to predict the binding affinity value of the PL complex are called as scoring functions[21]. Scoring functions are crucial for conducting and studying the Molecular docking and Molecular Dynamic simulation studies. The following are the major roles of SF's:

- Predict the absolute binding affinity of protein and ligand complex during lead optimization
- Predict binding site and mode of ligand to the protein
- Virtual screening i.e., identifying potential lead drug compounds from a large drug database that bind to target protein

Molecular docking is used to study how the two interacting molecules fit together as a bimolecular complex. Docking mechanisms use scoring functions to rank the best fit of the protein and ligand[22]. Scoring functions that are employed in molecular docking, such as AutoDock, X-Score, and ChemScore are based on the PL complexes which are semi-flexible resulting in lower computational accuracy. Also, docking can be very computer-intensive to carry out the screening of thousands of molecules corresponding to their targets. On the other hand, MD (Molecular Dynamics) Simulations make use of scoring functions such as MMPBSA (Molecular Mechanics Poisson-Boltzmann Surface Area), MMGBSA (Molecular Mechanics Generalized Born Sirface Area)[23], and Free energy perturbation. These scoring functions are based on flexible PL complexes and hence are more accurate yet computationally expensive. The mentioned scoring functions are considered traditional as these provide us with the theoretical basis for the prediction of BA and elicit recognition of candidate molecules.

The scoring functions are classified into four major types: *physics-based, empirical, knowledge-based, and machine learning-based*[23]. The first three types are based on linear regression-based methods and are differentiated on the basis of atom-type features. Also, these functions are based upon incomplete physical models which have been approximated to simplify the computation. They also involve complex operations and planning therefore, it is challenging for these methods to predict the BA accurately.
However, the fourth function, the ML-based function[24], uses non-linear regression models and statistical methods to train models and make predictions. These are much more robust and accurate than the other three scoring functions.

LITERATURE REVIEW

In order for a lead molecule to be classified as a drug, it must be able to bind tightly and specifically to the receptor protein, i.e., it must have a high binding affinity. The binding affinity value is of utmost importance in selecting lead compounds in the virtual screening stages of drug development and discovery. As the introduction section mentions, binding affinity is a quantitative value determining the attraction between the ligand(drug molecule) and the protein receptor(target molecule). This value is represented in terms of the dissociation constant (Kd) or inhibition constant (Ki). The dissociation constant, or Kd, can describe the affinity between the protein and the ligand molecules. The smaller the value of Kd, the stronger the binding between the protein and the ligand. In the case of enzymes and their inhibitors, the term Ki represents the binding affinity. The prediction of the binding affinity is an open challenge in the field of computational chemistry. Conventional methods used in binding affinity calculations, aka scoring functions(such as physics-based, knowledge-based, and empirical), are usually experimental and time-consuming.
In contrast, newer methods based on ML and DL are much more efficient and fast[25]. Several high-quality databases have been developed in the last few years to record experimental data on protein-ligand complexes. These databases can be used to develop Machine Learning and Deep Learning models to predict binding affinity. The performance of of the ML and DL models depends on the quality of data that has been used to design these models, hence it is emphasized that the datasets must be inclusive of high-resolution structures of PL complexes so that efficient ML/DL models can be developed[22]. Some of the major datasets and databases that can be used in the PL binding affinity predictions have been mentioned followed by published ML/DL models.

1. Protein-ligand Datasets

In this section, we discuss the most popular datasets that can be employed in developing, training, and evaluating the ML and DL models to predict the binding affinities. The following are the most popular datasets:

1.1. PDBbind

The PDBbind dataset was developed by Prof. Wang's group at the University of Michigan, U.S.A. It was released in the year 2004[26]. It is updated annually and is curated from the PDB(Protein Data Bank). It is a comprehensive collection of all the biomolecular complexes stored in the PDB. It consists of structural and energetic information on the biomolecular complexes that can be used in various tasks, such as

computational studies on molecular recognition, drug discovery, etc. It is one of the most used datasets in developing ML and DL models to predict the BA. The binding data for the molecular complexes is in the form of Kd, Ki, or IC50. The dataset is split into three parts, i.e., the general set, the refined set, and the core set[27]. These will be discussed further in the methodology section. The dataset can be downloaded from *pdbbind.org.cn.* This dataset has been used in our study extensively.

1.2. Binding MOAD

The Binding MOAD (Mother of all Databases) is curated from the PDB. It was introduced in the year 2005 and contained 5331 protein-ligand structures[28]. Currently, it has 41409 PL structures. It is a collection of biologically relevant and high-quality structures of protein-ligand complexes along with the binding affinities which are experimentally derived. The ligand molecules in the Binding MOAD consist of small peptides and oligonucleotides, drug-like molecules, and enzymatic cofactors. The resolution of crystal structures is better than 2.5 Å. The database can be found at *bindingmoad.org.* The Binding MOAD database and PDBbind database have been constructed in a similar fashion. However, the Binding MOAD consists of PL complexes along with their binding co-factors, complexes wherein both the ligand and the co-factor are present unlike the PDBbind dataset which has only the protein-ligand information.

1.3. CSAR

CSAR stands for Community Structure-Activity Resource. It is a collection of high-quality data from both Industry and academia. It is aimed at improving the docking scoring functions and also providing assessment methods for various PL binding affinity predictions[29]. The initial CSAR dataset consisted of the PL complexes from the PDB, for which the binding affinity data was available from the Binding MOAD.

1.4. Astex Diverse Set

The Astext Diverse Set is a dataset used to validate various ML and DL models, which are developed to predict the binding affinity. This dataset is composed of 85 protein-ligand complexes[30]. The dataset was curated as follows. Firstly, the proteins from the PDB were clustered together on the basis of their sequence similarity. This clustering led to 9188 clusters of proteins. Then, according to the drug-likeness criteria, the ligands bound to the proteins were clustered, which were of pharmaceutical and agricultural importance. The PL complexes were further assessed on the basis of

quality levels to obtain 427 clusters with high-quality PL data. The final Astex set was manually created to select 85 PL complexes.

2. Machine Learning approaches for prediction of protein-ligand binding affinity.

Machine Learning, aka the descriptor-based approaches, have been developed in the past several years to predict the BA. These approaches are data-driven, i.e., they involve using quantitative representations of molecules known as molecular descriptors as features. The simplest form of ML-based models are known as "QSAR" or Quantitative Structure Activity Relationships. These models consisted of molecular descriptors and multiple linear regression models[31]. Later on, models such as Support Vector Machines(SVM), Random Forests(RF), and Gradient boosting algorithms came into the ML landscape. These models aimed at development of non-linear relationships between the dependent variable i.e., the binding affinity, and the independent variables i.e., the descriptors.

One of the earliest ML models for BA affinity predictions combined the occurrences of protein-ligand atom pairs and distance-dependent atom-pair features with a K-PLS method(Kernel partial least squares) to predict the dissociation constant, Kd. This method showed that ML regression in combination with descriptors can be used to effectively predict the BA of PL complexes[32]. Later on, an SVM-based model was developed which involved the usage of three types of descriptors as features, (i). Property encoded shape distribution signatures, (ii). Molecular shapes encoding descriptors, and (iii). Property distribution on protein-ligand surfaces[33]. Another type of SVM-based model was developed, which used SVR-KB(knowledge-based) and SVR-EP(based on the physiochemical property of PL complex). Both these showed good results on the CASF benchmark dataset[34].

Random forest-based models have also been used to predict the BA. These have improved metrics and performance as compared to the SVM-based models due to the ensemble nature of the random forest algorithm. A novel random forest model, named RF-Score, was introduced[35]. The PL complexes were represented by a 36-dimensional vector for storing the features that stored the count of PL atom pairs within a predefined cutoff value of 12 Å. The feature vector was used as an input for the model. The RF-Score model can easily be compared to the other 16 types of scoring functions using the PDBbind benchmark. An updated version of the RF-score, called SFCScoreRF[36], was able to show improved performance as it used a larger feature vector to represent the PL complexes as compared to the traditional RF-score.

Another popular ML-based technique is based on the Gradient boosting algorithm combined along with decision trees. There are open source implementations present for the same known as LightGBM and Xgboost. Some of the ML based approaches for prediction of PL BA are known as XGB-Score and AGL-Score[37].
Another approach, improved upon the existing AutoDock Vina scoring function. It involved usage of a random forest based algorithm to correct the AutoDock Vina scoring function. This function is known as ΔVinaRF[36]. It shows good performance on the CASF 2016 benchmark dataset and also can be used in the virtual screening and docking studies. ML-based prediction techniques are constantly under development and improvement. One of the studies mentioned that, inclusion of RDkit based descriptors of the ligands may improve the BA prediction accuracy[38]. Several other techniques have been developed that are based on use of ML in BA prediction. These make use of kernel ridge regression and Gaussian processes[39]. From the above account of the various ML techniques that have been developed for the prediction of BA, we observe that there is still scope for improvement as these methods may perform poorly in the novel virtual screening taks that are to be implemented during the drug dicovery process. Hence, proper training and data processing becomed empirical in such techniques. We have discussed many DL techniques in the next section that have shown significant improvements in the accuracy of the BA task as compared to ML-based binding affinity prediction techniques.

3. Deep Learning approaches for prediction of protein-ligand binding affinity

Initally, the neural network consisted of single layer of neurons connected to the input and output. However, with the progression in the Deep Learning landscape, the neural networks were improved upon by the introduction of multiple layers of neurons in the hidden layers and the network architecture become deeper and complex[40]. These multi-layer neurons came to known as Deep Neural Networks, or simple DNNs.

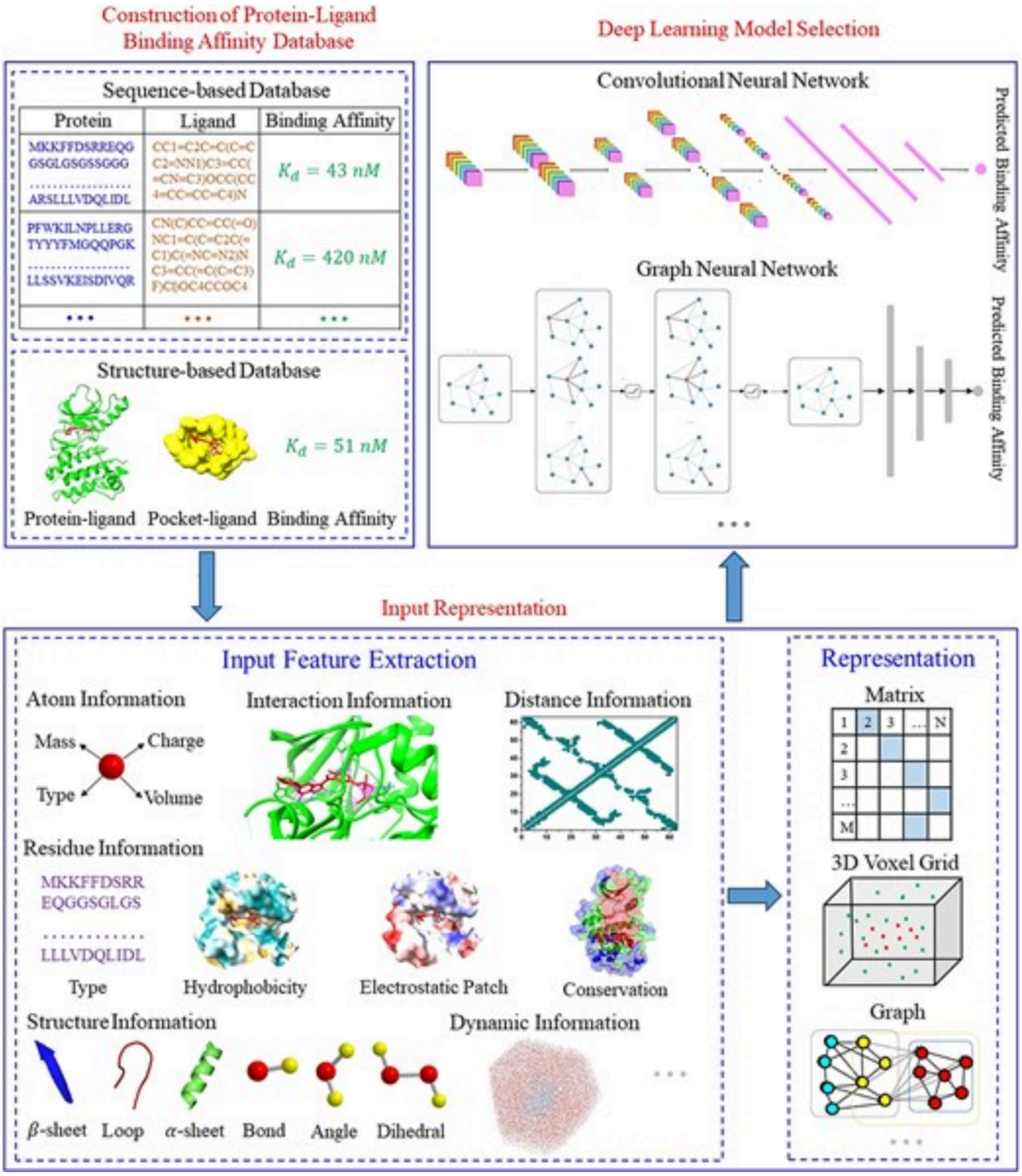

Figure 4: Overall flowchart showing DL based approaches to predict BA[40]

Neural networks:

The neural networks have been used for establishing the QSAR models[41]. A neural network was developed which was used for classification tasks and for differentiating between the actives and decoys. This type of neural network model came to be known as NNScore[42]. It was able to differentiate between well and poorly docked ligands along with the actives from the decoy molecules. NNScore was later extended into a regression model known as NNScore 2.0, which allowed for estimation of the dissociation constant value represented by the term, p(Kd)[43]. This model makes use of the Vina scoring function so as to encode the interactions such as steric, hydrophobic

and H-bonding along with the BINANA features. These were used as input to the neural network architecture which returned the estimated p(Kd) value.

CNNs:

Convolutional Neural Networks abbreviated as ConvNets or CNNs, have been employed to overcome the limitations of the comparatively simpler DNNs. The CNN architecture consists of input/output neurons, hidden layers, convolutional layers and fully connected layers. The traditional feed-forwards neural networks or the DNNs make use of one-dimensional vectors to explain the protein-ligand complexes which were not able to incorporate the 2-D, 3-D or the spatial characteristics of the input structures[44]. Models such as AtomNet[45] and KDeep[46] were based on CNNs and used vectorized grid into a cubic box of pre-defined dimensions which was centered at the ligand center for computation of the protein-ligand feature extraction process. Using CASF-2013 dataset as the benchmark set, a CNN model known as Pafnucy[47] has been a very crucial CNN model developed so far that showed good performance(R = 0.7). In the Pafnucy model architecture, the grid box (20 Å × 20 Å × 20 Å) was centered at all the ligand atoms and was extracted at every 1 Å grid. The resultant dimensions of the dataset were 21 × 21 × 21 × 19. This was fed into the CNN model so as to make predictions. We have benchmarked this model as a part of our comparative study to develop and benchmarks the ML and DL based model to predict the binding affinity values. Another popular model known as OnionNet[48] was developed based on the CNN architecture. It is a multi-layered model, based on the intermolecular contacts between the ligand atoms and the protein residues. This model takes into account, both local and the non-local interactions between the protein and the ligand atoms and groups them into the various distance based shells or ranges. Each shell or the distance range accounts for eight atom types which makes up for 64 (8 ligand atom types X 8 protein atom types) features in each distance shell. A total of 60 shells were employed with a thickness of 0.5 Å each. Another CNN-based model known as OnionNet-2[49], extended the approach followed in the OnionNet model. This model used protein atom residues instead of the atom types in the proteins which were used in the OnionNet model. This increased the features form 64 to 168(8 X 21). This model has also been benchmarked as aprt of our study. Another CNN-based model with lower-dimensional descriptors known as TopologyNet[50] was introduced. It encodes the 3-D structure of the PL complex to the 1-D element specific fingerprints. These elements are then stacked upon one another to form a 1-D representation of an image and fed into the CNN model. CNNs have changed the landscape of the models that have been developed so as to predict the PL BA. However, CNN models are translatioanlly invariant and require a lot a data augmentation. This is due to the fact that the PL

complex is randomly rotated before being used to compute the grid. Hence, these models cannot be considered rotationally invariant.

GNNs:

GNNs or the Graph Neural Networks are a class of Deep Learning models which are constructed so as to represent the input data in the form of a graph with nodes and edges like representation. The molecules can be represented in the form of graphs, wherein the nodes represent the atoms and the edges represent the various chemical bonds between atoms. Since, the molecules can be simply described by graphs, GNNs are very relevant in the filed of computational chemistry[51]. A model named GraphDelta[52] was developed in which the ligand was described with the help of a graph based representation and the protein target information was incorporated into the graph via the edges. After training for a sufficient number of epochs, the model achieved a Pearson correlation coefficient of 0.87. Another model named GraphBAR[53], was developed in which a graph is constructed with all the protein and ligand atoms within a specific range of 1 Å of the ligand molecule. A GNN based model known as InteractionGraphNet[54], was developed where two graphs which are independent to one another were stacked on each other. This allowed for sequential learning of intermolecular and intramolecular interactions using three molecular graphs. One graph represented the protein, the other represented the ligand and the third one represented the PL complex. The use of GNNs have allowed for geometric representations of the PL complexes making this approach a very functional approach towards the virtual screening process using DL models.

CHAPTER 1 :: Data-driven Descriptor based approach

1.1. Introduction:

1.1.1. Regression: A supervised ML technique

This chapter focuses on a machine learning-based approach to predict the PL binding affinity based on the fundamental molecular properties represented as descriptors. ML models can be used to train the inputs to make data-driven predictions which are finally expressed as outputs. These methods can statistically learn the correlation between chemical structures representations and the output status of the known PL complexes so as to predict the same for the unknown. Machine learning algorithms can either be supervised or unsupervised. In this study, we will be focussing on the supervised aspect of the ML algorithms known as regression. Regression is a set of statistical methods which can be used for estimating the relationship between the dependent continuous variable and one or more independent variables[55]. Our aim is to predict the binding affinity value which is continuous in nature hence we will be using regression based Machine learning algorithms. The main task for the regression algorithm is to find out the best fit line equation that can predict the dependent variables based on the independent variables. The best fit line implies that the difference between the predicted and the actual values of the dependent variables are kept to a minimum.

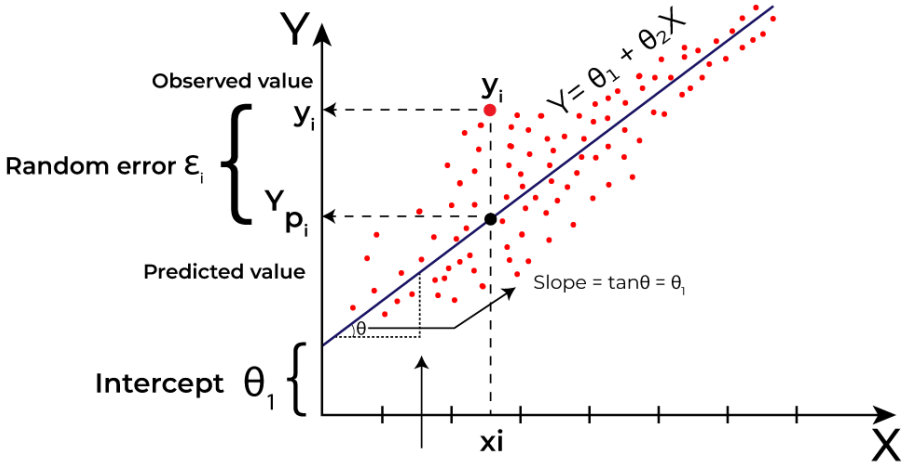

Figure 5 : Scatter plot showing the best fit line in linear regression.

In the figure shown above, Y is the dependent variable, and X is the independent variable. X may represent a single feature or multiple features depending upon the problem we are dealing with. If we assume that there is a linear relationship between X and Y, then the binding affinity can be predicted using the equation of a straight line which is given alongside the best-fit line in the figure. The model gets the best regression fit line by finding the best θ1 and θ2 values, where θ1 represents the intercept and θ2 represents the coefficient of X. It is crucial to update the θ1 and θ2 values so as to reach the best-fit line which minimizes the error i.e., the difference between the actual and the predicted values[56].

1.1.2. Evaluation metrics for regression

The cost function aka the loss function can be defined as the difference between the actual and predicted dependent variables[57]. It is also known as the error or residual error. Some of the most common cost functions employed in the regression tasks are mentioned below:

- MSE (Mean Squared Error) : It is generally used to quantify the accuracy of models based on linear regression. It is used to calculated the average of squared differences/ errors between actual and predicted dependent values. The MSE is used to optimize the value of the intercept θ1 and the coefficient of of the input feature i.e., θ2 eventually providing the best fit line. So as to optimize the mentioned values the process of gradient descent is employed which is iterative

in nature. This allows for the MSE values to converge to a global minimum. MSE is sensitive to outlier values as large error values affect the model performance significantly.

- MAE (Mean Absolute Errror) : MAE is used to measure the average of absolute errors between the actual and the predicted values. Lower the MAE, better the model performance. Also, in contrast to MSE, the MAE metric is not sensitive to outliers.
- RMSE (Root Mean Squared Error) : RMSE is calculated by taking the square o the variance of the residual errors. It represents the model's absolute fit to the actual data. It is not a normalized metric, as the value of RMSE can fluctuate when the dimensional units of the independent variables vary. This is because, its value is dependent on the units of the variables.
- R-squared: It is a metric which captures the variation that the model can explain. It ranges from 0 to 1. Generally, if the model fits to the data properly then the R-squared is greater. If the value of R-squared is closer to 1, then the model is said to be preforming better.

1.1.3. Molecular Descriptors:

Molecular descriptors can be defined as the quantitative representations of the properties of molecules that are generated byu mathematical functions and algorithms. The numerical representations are utilized to describe the physical and chemical properties of molecules among others[58]. The most commonly used molecular descriptor in the field of chemoinformatics is calles LogP, which is a value that represents the lipophilicity of the molecules[59]. Experimentally, it is obtained by measurement of the partitioning of a molecule between two phases i.e., an aqueous phase and a lipphilic phase which contains water/n-octanol. A wide variety of molecular descriptors have been implemented for research tasks for drug discovery in the past few years. Some examples of molecular descriptors include molecular weight, number of carbon atoms, properties computed from 2D structures such as the Eccentric Connectivity Index, and 3D structures such as charged partial surface area(CPSA). Other type of descriptors derived from quantum mechanical properties of compounds are of interest to researchers. These include properties such as HUMO and LUMO[60]. The descriptors are crucial to describe the properties of molecules of interests such as ligands, small molecules, proteins etc., so as to be fed into Machine Learning models as features. These features in turn can help train models so as to predict the dependent variables, either in classification or regression tasks. Also, the choice of descriptors plays a avery important role while developing the models as the predictive power and accuracy of the model depends on the same. Useful features can be selected by employing feature selection tasks. In this work, we have used the molecular descriptors so as to build ML models which can be used to predict the binding affinity values of the

protein-ligand complexes. The molecular descriptors can be computed using a variety of libraries and tools. For instance, for computing the molecular descriptors of ligands, we can make use of tools such as Rdkit, Mordred or Padel[61]. For sequential inputs such as proteins, we can use tools such as Pfeature.

1.2. Materials:

1.2.1. Dataset: PDBbind

PDBbind is a collection of experimentally derived binding affinity data for the molecular complexes that are present in the PDB(Protein Data Bank). The database was developed by Prof. Shaomeng Wang's group at the University of Michigan, USA. Some of the applications of the database include study of molecular recognition between complexes and drug dicovery[26]. For this study, the current release of the database i.e., the PDBbind(version 2020) dataset released in the year 2020 was used. This dataset is based on the contents of the PDB released in the year 2020, which consisted of 157,974 experimental structures. The entire PDB was further screened to identify 78,460 PDB entries under valid complexes. Further, the primary reference of each complex which falls under the binding affinity being given as Kd, Ki or IC50 is selected. This gives a general set of 23,496 complexes. These valid bimolecular complexes include:

- Protein-ligand (19,443)
- Protein-Protein (2852)
- Protein-nucleic acid (1052)
- Nucleic acid-ligand (149)

Additionally, a refined set of 5,316 protein-ligand complexes depicting better quality out of the general set has been given. This subset of the data was used for the studies mentioned in this chapter. Here are a few important points to note about the refined set:

- The refined set was downloaded from the PDBbind website as a '_tar.gz' file and unzipped on the local machine.
- The folder consisted of 5318 subfolders. Out of which 5316 folders were protein-ligand complexes and the other two were additional supportive information including the binding data and readme documents.
- Each protein-ligand complex folder was named based on the protein PDB ID. For instance, the protein with ID, '1A1E' and its ligand were named as '1A1E'.
- Each folder consisted of a few structural files. These were:
    - Protein file in PDB format
    - Pocket file in PDB format
    - Ligand file in sdf format
    - Ligand file in mol2 format

1.2.2. Descriptor calculation tools/libraries

Protein Descriptors:

For the calculation of protein descriptors, PFeature tool was used. It was developed by Dr. Raghava's group at IIIT-Delhi[62]. This tool provides web server and command line options for computing protein features from amino acid sequences. There are four main types of features that can be computed using the tool, these are:

- Composition-based features
- Binary profiles of sequences
- Evolutionary information based features
- Structural features

Ligand Descriptors:

For the computation of ligand descriptors, two python-based libraries namely, RDkit[63] and Mordred[64] were used. RDkit is a chemoinformatics and ML software that is written in Python and C++. It is used for computing around ~300, 2D, and 3D features from molecules. Mordred is a molecular descriptor calculator software that can compute ~1800 2D, and 3D molecular descriptors including the ones that are computed by RDkit. It can be used both in command line modes and as a web server tool.

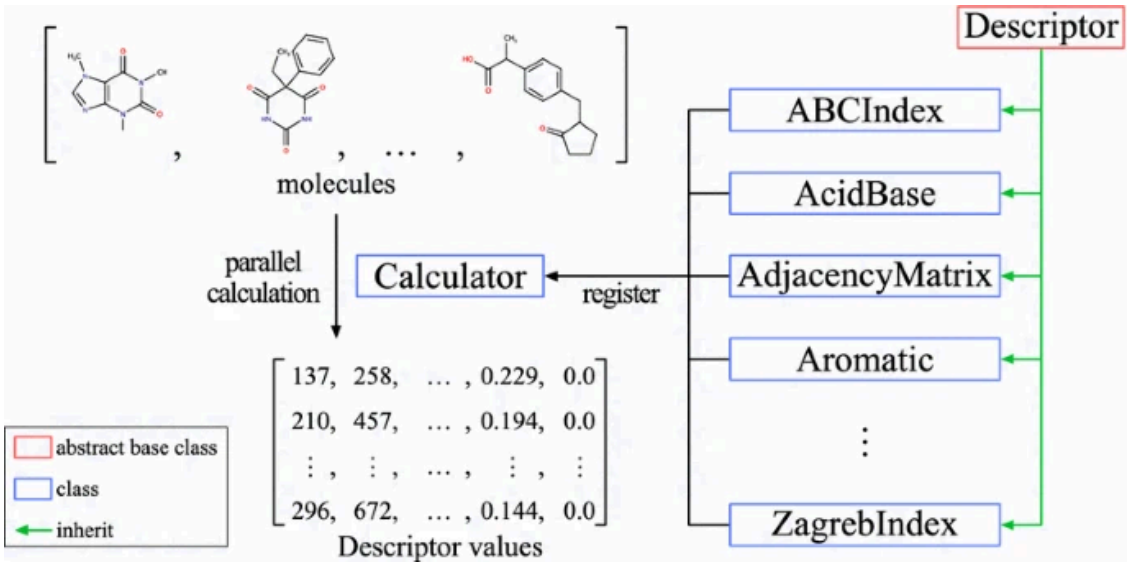

Figure 6 : Overview of Mordred library[64]

1.2.3. Python libraries

Python libraries are crucial to carrying out data analysis and processing tasks. The most important task is to handle the data efficiently, which can be done by implementing libraries such as Pandas and NumPy. Pandas is a powerful library that is flexible in nature and can be used to handle large amounts of tabular data along with data manipulation. NumPy is a scientific library that allows the user to carry out mathematical operations on multi-dimensional data such as arrays and matrices. It is also helpful to carry out linear algebra-based tasks. The second class of Python libraries used includes Scikit-learn. It is used for carrying out tasks such as data analysis, pre-processing, scaling, splitting model building and evaluation. The third class of libraries used involves data plotting and representation. For this, Matplotlib and Seaborn were used.

1.2.4. System Requirements

This part of the project was run on the local machine. The operating system used is MacOS Big Sur (version 11.1).
The programming language used is Python which is the most used language for data science and machine learning tasks. It is popular for its easy-to-understand syntax and readability. The prototyping is quick meaning that the code can be implemented as soon as it is written.
Jupyter Notebook is a versatile web-based computing and development platform. Its interactive interface allows users to manage their notebooks, code, and data efficiently. It provides a document-centric coding environment that finds its application in the field of Data Science and Machine Learning. Jupyter allows users to code in 40 different languages including the most popular ones i.e., Python, Scala, Ruby etc. Jupyter notebooks were extensively used in this study for coding and managing the data.

1.3. Methods

The following section will consist of the methodology adopted for the first approach for predicting the binding affinity of protein-ligand complexes using Machine Learning techniques.

1.3.1. Dataset preparation: Feature extraction

Binding Data Preparation:
The binding data was available as a part of the PDBbind refined set in the form of Ki/Kd/IC50in the various units fM, nM, pM. These were all brought to the same scale i.e., Molar and followed by conversion to the negative log base 10 of (IC50 or Ki or Kd), which gives p(IC50). This value was considered as the dependent variable.
Formula for conversion: *-log10(IC50 or Ki or Kd) = p(IC50) = Binding affinity*

Feature extraction:
The protein-ligand complex data was taken from the refined set of the PDBbind 2020 version as mentioned above. There were 5316 PL complexes. The ligand SDF files were used to compute the RDkit descriptors using the RDkit Python library. Consequently, the Mordred descriptors were computed from the ligand SDF files using the mordred library in Python. Both the descriptor sets were saved in separate csv files.
As for the protein features, the protein PDB files were first converted to their respective amino acid sequences as a dataframe using the PDBParser function of the BioPython module available in Python. Then, the protein with non-natural amino acids were excluded from the dataframe. Now, the amino acid sequences were used so as to compute the protein features from PFeature. The standalone version of the PFeature was downloaded from the GitHub repository. Some of the features computed using Pfeature include, AAC (Amino Acid Composition), DPC (Dipeptide Composition), ATC (Atomic Composition), BTC (Bond Composition), PCP (Physio-Chemical Properties) and SEP (Shannon Entropy of protein) among others. All the features were concatenated to form one single feature dataframe.
Finally, two final dataframes were formed, which represented two cases:

- Rdkit ligand features + Protein features (Case 1)
- Mordred ligand features + Protein features (Case 2)

1.3.2. Feature selection:

Since the features were in hundreds, there was a need to select features that were relevant in predicting the binding affinity values. These are a lot of techniques that can be employed for the same such as correlation, mutual information gain, etc. For our case, we used the correlation coefficient-based approach. The collinear features were removed which had a correlation coefficient greater than a certain threshold. The threshold taken was 0.9. Removing collinear features helps the model to generalize and improve its interpretability.

1.3.3. Splitting and Scaling the Dataset:

The dataset was split into a ratio of 70:30 for Train and Test data. This was done using the train_test_split function of the sklearn.model_selection library. This function splits the dataset into four parts, these are mentioned below:

- X_train : Includes the training features
- y_train : Includes the training labels/predictors/dependent variables
- X_test : Includes the test features
- y_test : Includes the test labels/predictors/dependent variables

The dataset was scaled using the StandardScaler function of the sklearn.preprocessing library. Standardization is done so as to rescale the mean of the data to 0, and the standard deviation to 1. This measure assumes that the input data fits a Gaussian distribution.

1.3.4. Model Building:

Various regression models were used for our study including Multiple Linear Regression, Support Vector Regressor, Random Forest Regressor, Gradient Boosting Regressor etc. These models were employed using the scikit-learn library. Each model requires a model object to be initialized followed by fitting the model to the training set. Finally, the predictions are made on the test set. Here, the predictor variables are the binding affinity values.

1.3.5. Evaluation metrics:

The models are evaluated in two ways, quantitatively and graphically. Quantitative evaluation involves looking into metrics specific to regression, such as R-squared, MAE, MSE, RMSE. These are interpreted according to the values that are predicted i.e., binding affinity values.The metrics are computed using the metrics class of the sklearn library. Graphically, the models are evaluated by studying the scatter diagrams plotted between the actual and predicted values and the best fit line. The graphs are formed using libraries such as Matplotlib/Seaborn.

1.4. Results

CASE 1: Results obtained for RDKit features+Protein features

1.4.1. Metrics for regressor models employed for predicting binding affinity

Model	R_squared	MAE	MSE	RMSE
Multiple Linear Regression	0.44	1.21	2.25	1.50
SVM regressor	0.58	1.03	1.70	1.30
Random Forest regressor	0.60	0.98	1.60	1.26
Gradient Boosting Regressor	0.56	1.06	1.77	1.33
Bagging Regressor	0.57	1.02	1.74	1.32

Table 1: Evaluation of Machine Learning model metrics for predicting PL BA

1.4.2. Scatter diagrams for the actual v/s predicted binding affinity values

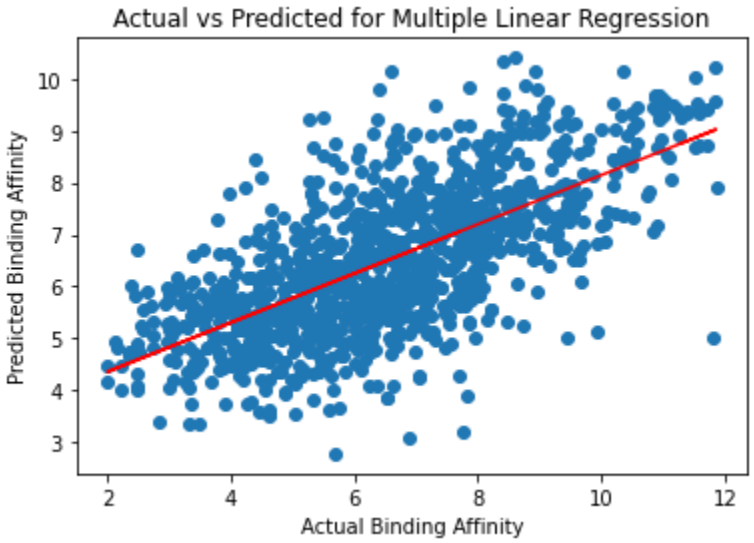

Figure 7: Scatter diagram showing the actual v/s predicted BA using MLR

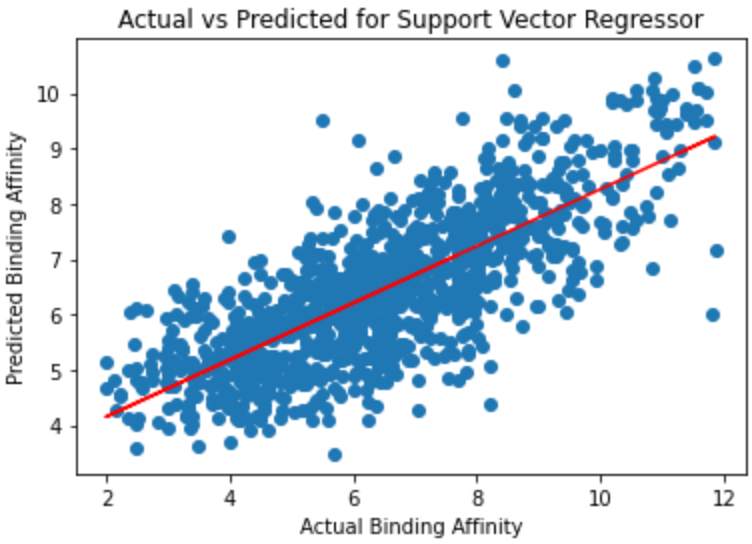

Figure 8: Scatter diagram showing the actual v/s predicted BA using SVR

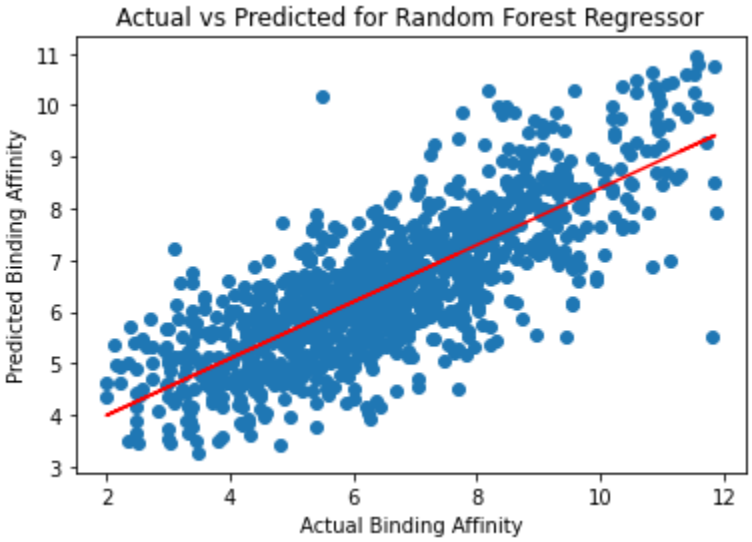

Figure 9: Scatter diagram showing the actual v/s predicted BA using Random Forest

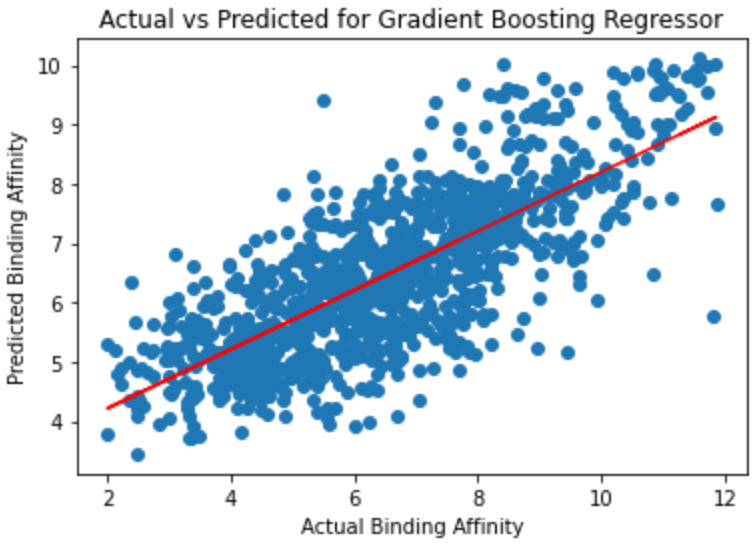

Figure 10: Scatter diagram showing the actual v/s predicted BA using Gradient Boosting Regressor

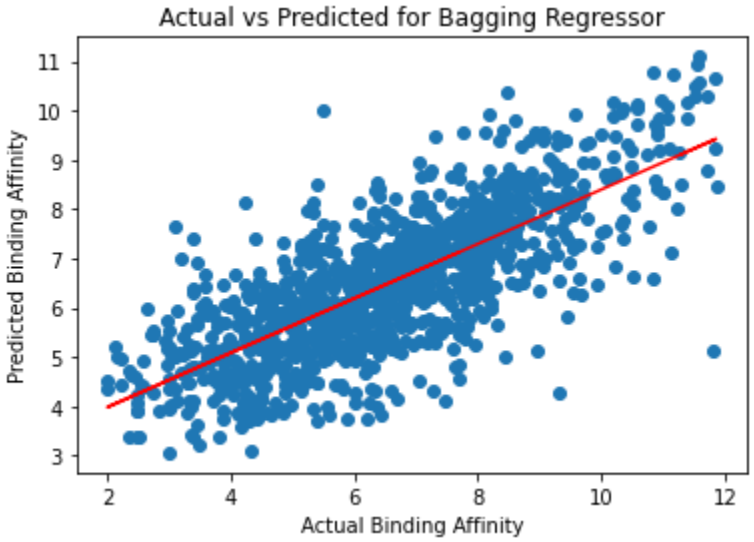

Figure 11: Scatter diagram showing the actual v/s predicted BA using Bagging Regressor

CASE 2: Results obtained for Mordred features+Protein features

In the second case, the protein features were computed using the Pfeature tool and the ligand features were computed using the Mordred tool. This case gave results that were quite similar to the first one. The Random Forest regressor showed the best metrics, i.e., R-squared ~0.60, MAE of 1.00, MSE of 1.62 and RMSE of 1.27. The scatter

diagram for the Random Forest Regressor model showing the actual vs predicted binding affinity values has been given below.

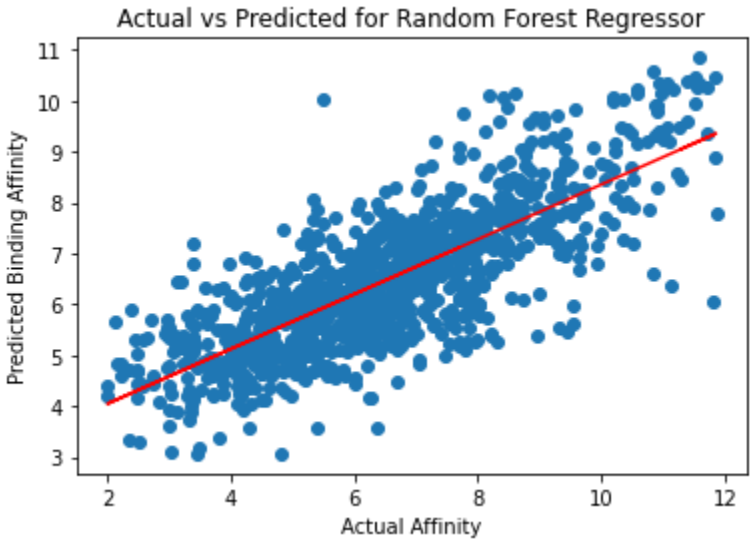

Figure 12: Scatter diagram showing the actual v/s predicted BA using RF

1.5. Discussion

Although we tried a number of feature selection techniques on the dataset, the best results were obtained from using the correlation-based technique. Also, out of the models tested, the best models that performed for the regression task have been mentioned in the results section. Their scatter plots show the variation of the actual values from the predicted values. The Random Forest model outperformed all the models in both cases. Random Forest is an ensemble-based decision tree model. The performance of the model could be attributed to the following reasons:

- The Random Forest algorithm operates by constructing a number of decision trees during training and outputs the mean prediction of the individual decision trees.
- It combines the result of a number of trees i.e., it is a meta-estimator.

CHAPTER 2 :: Benchmarking the Pafnucy model

2.1. Introduction

2.1.1. A brief intro to Deep Learning

Deep Learning, abbreviated as DL is a branch of Machine Learning and it is considered as one of the core parts of the Fourth Revolution technologies (4IR or Industry 4.0). Artificial Intelligence incorporates intelligence to machines. DL represents learning from data using multiple layers of Neural Networks. Neural Network or NN, is a generic term given to the computational model which is inspired by the working of the neurons in the human brain. NNs are composed of interconnected structures of nodes aka neurons, Neurons in a neural net are responsible for taking input, processing the input, and giving out output. Hence, forming the three most crucial layers, i.e., the input layer, hidden layer, and output layer. Some of the other essential components of a neural network include the summation function, activation functions, and weights.
DL has exceptional learning capabilities which are derived from input data. Some of the most popular use cases and applications include healthcare, text analytics (Natural Language Processing), visual analytics and biology domains such as drug discovery etc[44].

2.1.2. DL Techniques

There are three major categories of DL techniques, i.e.,

- Supervised/Discriminative Learning is used in classification tasks and makes use of labeled data. The DL techniques used in Supervised learning include MLP (Multi-Layer Perceptron), CNN (Convolution Neural Networks) and RNN (Recurrent Neural Networks) which further can be divided into LSTMs (Long Short Term Memory).
- Unsupervised/Generative Learning is used in pattern analysis and characterization of features and makes use of unlabeled data. Examples include Autoencoders, GAN (Generative Adversarial Networks).
- Hybrid Learning is a combination of both the above-mentioned approaches. Examples include AE+GAN, an integrative model of generative and discriminative models.

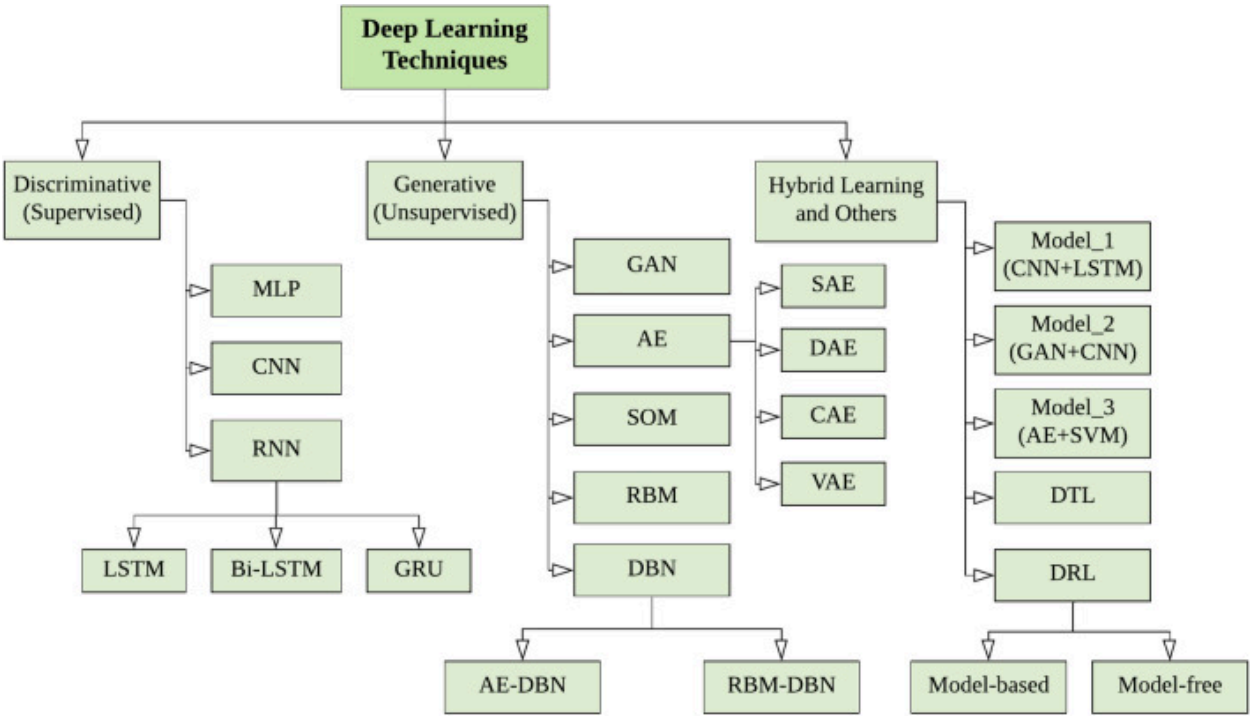

Figure 13: Categories of Deep Learning[65]

2.1.3. Forms of Data fed into DL models

DL models make use of data given by the user to learn and understand the data so as to build an intelligent system to carry out the desired task. The data fed into the DL models can be of various types, these have been described below[65]:

- Tabular Data
  It consists of rows and columns where each row specifies an entry/sample and the column represents its characteristic feature. Each column must possess data of the same type.
- Sequential Data
  The type of data where order is crucial i.e., a set of DNA/RNA sequences. Each data point in the sequence is dependent on the other data point. Some common examples include time series data, genomic data, etc. Some of the important characteristics of an image include pixels, voxels, matric, and bit depth.
- Image Data/ 2D data
  An image is basically composed of a matrix-like structure including representative numbers, and expressions that are arranged in 2D arrays.

2.1.4. Deep Networks for Supervised Learning: CNN (The scope of this study)

The CNN or the ConvNet is a type of supervised or discriminative DL technique that is capable of learning from the input data without the requirement of pre-processing stages such as feature extraction. It is used mostly in the cases where visual images are involved. It has a very crucial role in the field of pattern recognition making it one of the widely used techniques in computer vision applications[66]. There are a number of layers that make up a CNN model, which have been described briefly below:

- Convolutional Layers: These are tasked at applying convolutional operations on the input images (aka kernels) so as to detect features such as edges, textures etc. These layers are important for the preservation of spatial relationships between pixels.
- Pooling Layer: These layers reduce the complexity of the input data by downsizing the spatial dimensions of the input imagery data. It involves the selection of maximum values from a set of pixels.
- Activation Functions: These functions are used to introduce non-linearity into the model. Some functions include ReLu, sigmoid etc.
- Fully Connected Layers: These are the final layers of the CNN model responsible for making the predictions based on all the above layers.

2.1.5. *Pafnucy* : A CNN-based model to predict the PL Binding Affinity

Pafnucy, is a deep neural network that is used as a virtual screening model to carry out structure-based tasks specifically relating to the prediction of binding affinity[47]. The input in this model is represented using a 3-dimensional grid form. It involved convolutional and dense layers so as to predict the BA. It also involves crucial steps such as feature extraction and importance. This model approaches the protein and ligand atoms in a similar way, i.e., it does not distinguish between the two. The model allows accounts for the PL interactions by serving as a regularization technique. In this chapter, we focus on benchmarking the model so as to make predictions on our own test PL complex dataset. This eventually allows us to test the model on an unknown dataset and could help with future drug discovery tasks such as virtual screening by predicting the essential property of PL complexes i.e., binding affinity.

2.1.6. Features and dataset preparation for the Pafnucy model

So as to be utilized by a neural network, the three-dimensional structures of the protein and ligand complexes need to be transformed and encoded to be fed into a CNN model.

Feature Extraction:

In the Pafnucy model, the PL complex is cropped to a size of 20Å cubic box which is focused at the center of the ligand molecule. The heavy atoms in the cubic box were further discretized into 3-D grids of sub 1Å grids. This step allowed the input to be represented as a 4-D tensor-like structure that can be further defined by cartesian coordinates and characteristic features. The cartesian coordinates represent the first three features, and the other features are considered as the last dimension.
There are 19 features that represent the PL complex in this model, which are given in a tabular format as follows:

S.no	**Feature Type**	**Bits encoding**	**Feature representation**
1	Atom Types (computed with Openbabel)	9 (One hot or all null)	*B, C, N, O, P, S, Se, halogen, and metal*
2	Atom Hybridization	1 integer (1,2, or 3)	*hyb*
3	Number of bonds with other heavy atoms	1 integer	*heavy_valence*
4	Number of bonds with other heteroatoms	1 integer	*hetero_valence*
5	Properties defined with SMARTS patterns	5 bits (1 if present)	*Hydrophobic, aromatic, acceptor, donor, and ring*
6	Partial charge	1 float	*partialcharge*
7	Protein or Ligand	1 integer (1 for ligand and -1 for protein)	*moltype*
Total Features:		**19**	

Table 2: Features representing the PL complex in the Pafnucy model

Dataset Preparation:

The PDBbind dataset, version 2016 was used for the training of the model. The protein-ligand complexes were extracted and protonated and charged using the UCSF Chimera. For the validation set, 1000 PL complexes were selected randomly from the refined set. The core set was used as a test set. The remaining refined set PL complexes and all the general set complexes were used for training purposes.

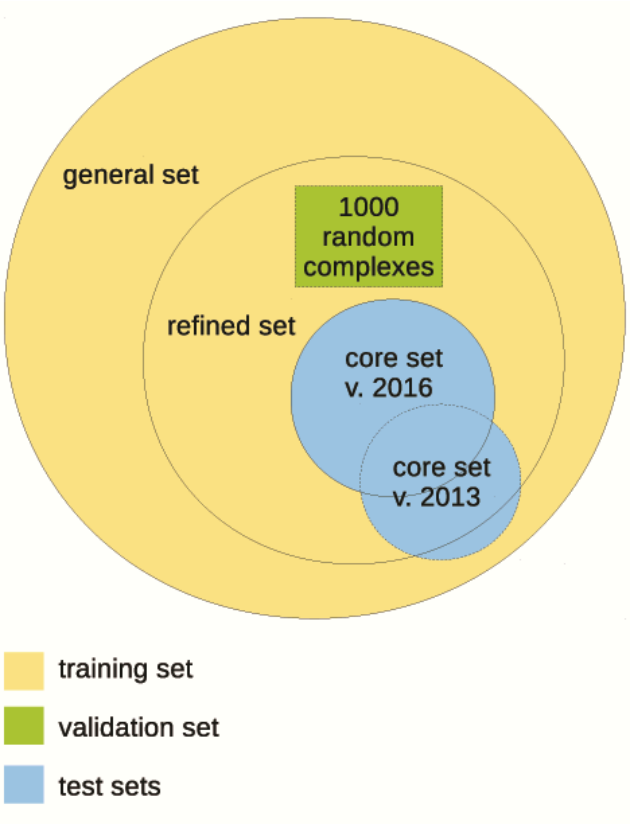

Figure 14: Dataset preparation and division (Pafnucy)

2.1.7. Pafnucy Model Architecture and Results

Architecture:

The input in the model, which is a molecular complex, is represented by a 4D tensor and treated as a 3D image. The 3D image has multiple color channels. Each input position (the cartesian coordinates) is described a vector having 19 features. This is analogous to the pixel of an image where the x, y, and z coordinates basically imply the three RGB colors in a pixel.

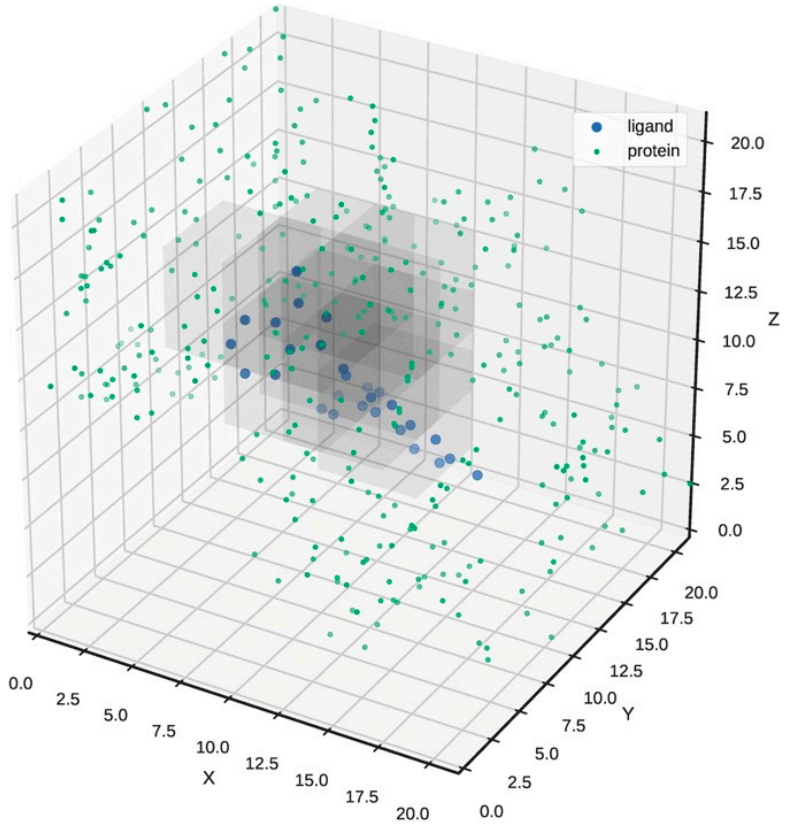

Figure 15: The model architecture showing the 4D tensors and the orientation of the grid

The model makes use of CNN model consisting of a single output neuron to predict the binding affinity of a PL complex. The model is composed of two parts, i.e.,

- Convolutional Layers
  Convolution implies the mixing of functions. The model basically discovers the patterns encoded by the convolution layer filters and generates filter maps so as to represent the spatial characteristics. The input is processed by the segment of

3D Conv layer with 64, 128 and 256 filters. Each layer has 5-Å cubic filters and is followed by a max pooling layer with a 2-Å cubic patch.

- Dense Layers
  The result of the last convolutional layer is flattened and used as input for a block of dense (fully-connected) layers. The model uses three dense layers with 1000, 500 and 200 neurons.

An activation function known as ReLU or the rectified linear unit is used in the composition of both the layers mentioned above. ReLU is used to expedite the learning process.

Metrics for the Model:

Dataset	**R**	**RMSE**	**MAE**
Test (v. 2016 core set)	0.78	1.42	1.13
Validation Set	0.72	1.44	1.14

Table 3: Pafnucy model performance according to paper[47]

The features that described the model best were also extracted using the analysis of the distribution of weights of convolution filters in the first layer of the model. This could be done due to the fact that training was done using L2 regularization. It was found that moltype feature was giving the best performance as it describes the relationship between two molecules.

2.2. Materials

2.2.1. Dataset

The refined set of the PDBbind dataset version 2020 was used in this part of the study. The details of which have been mentioned previously. All the complexes in the refined set were used to carry out the benchmarking task. The model takes the protein pocket and the ligand structures as input, so these files were extracted from the refined set. The protein files were in PDB format, while the ligand files were in mol2 or SDF format.

2.2.2. Files pre-processing: OpenBabel

OpenBabel[67] is a chemical toolbox that allows working with chemical data so as to convert, analyze, search, manipulate, and process data. It is a significant tool used in the field of computational chemistry. It supports many file formats including the most popular ones i.e., PDB, SDF, mol2 etc. It was downloaded and can be used in any directory by suing "obabel" as a prefix in the command. Openbabel was used to protonate and charge the PL complexes.

2.2.3. Pafnucy command-line software

The command line version of the Pafnucy was cloned from the GitHub repository. The command line version is developed using Python programming language. Firstly, A designated Python environment named *pafnucy_env* was created and all the required dependencies were installed in the environment. This was followed by using the command further to carry out the prediction tasks. The model predicts the binding affinity in terms of -log10(IC50).

2.2.4. System Requirements

The repository was downloaded on the department's Linux x86_64 architecture server. Miniconda, which is a small bootstrap version of Anaconda, was installed on the server. It is a freely available minimal installer for conda. It includes conda, Python, and packages that both depend on. For installing additonal packages the command *pip install* is used. Due to the large number of PL complexes to be worked upon, it was rudimentary to use the sever instead of local machine due to limited computational power of the local machine.

2.3. Methods

2.3.1. Dataset preparation

The protein pocket and ligand structural files were separated from the refined set of the PDBbind 2020 version. The protein pocket files were in PDB format and ligand files were in mol2 format. Further, the PL complexes were protonated and charged using the

open babel command line toolbox. The commands to carry out the same are mentioned as follows:

General command:
obabel <input file path> -O <output file ID> -<task>

For protonating and charging:
obabel <input file path> -O <output file ID> -partialcharge -h

2.3.2. Model Usage

The following steps were followed to benchmark the Pafnucy model to compute binding affinity values of protein-ligand complexes:

- We need to activate the Python environment namely, *pafnucy_env* to use the dependencies that the model requires. The environment is activated using the command : ***conda activate pafnucy_env***
- The next step is to prepare the complexes and the second step is to predict the binding affinity. Both these steps can be carried out using the commands given below:
  Preparation: ***python prepare.py -l ligand.mol2 -p pocket.mol2 -o data.hdf***
  Prediction: ***python predict.py -i data.hdf -o predictions.csv***
- The output is stored in a CSV file, which gives the complex PDB ID as first column and the binding affinity (Ki or Kd) as second entry.
- The above steps can only process one protein ligand complex at a time so, the task had to be iterated for all the PL complexes in the refined set that had to be tested. This was done using a Python script which made use of the *subprocess* module. It basically wraps codes with Python and allows for simplified and easier use of command-line scripting options. The commands given above could now be run using a Python program. All the PL complexes were iteratively processed using a for loop and the binding affinity prediction data was appended to a single CSV file.
- Finally, the CSV file had a PDB ID column and a prediction value column.

2.3.3. Evaluation of the model results

For comparing the actual and predicted values of BA, the units of the same had to be dealt with. The predicted values were in the form of p(IC50), so the actual values of PL binding affinities in the refined set were converted to same scale first. This was done by

first converting the values to Molar units and then to respective BA values similar to the predicted values scale. The conversion formula used is :

-log10(IC50 or Ki or Kd) = p(IC50) = Binding affinity

For evaluation of the model, regression metrics such as R-squared value, MAE and RMSE were computed using the sklearn library. This was followed by plotting a scatter diagram for actual vs the predicted values of binding affinity.

2.4. Results

2.4.1. Test result metrics

The model tested on the refined set (PDBbind v. 2020) was evaluated quantatively by computing the following regression metrics.

Test Dataset	**R-squared**	**MAE**	**MSE**	**RMSE**
PDBbind refined set (v. 2020)	0.464	1.136	2.037	1.427

2.4.2. Test result scatter plot

The scatter diagram showing the plot of actual vs predicted binding affinites is as follows:

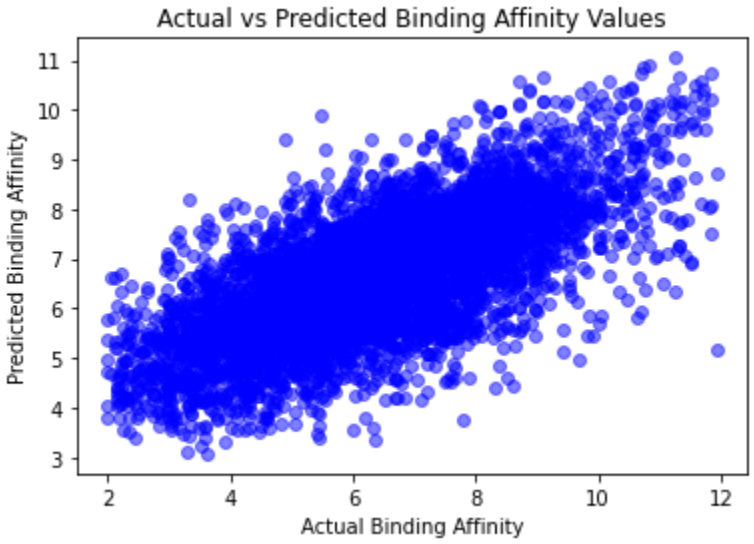

Figure 16: Scatter diagram showing the actual v/s predicted BA using Pafnucy

2.5. Discussion

Pafnucy is a notable tool which holds potential in the virtual screening process of the drug dicovery campaigns. It can be used to predict the binding affinity values for novel PL complexes. The model gave an R-squared value of ~0.46 when we tested it on the refined set of PDBbind (v. 2020) dataset. This result is comparatively lacking as compared to the results of the initial studies that were performed in the paper. This could be due to the fact that deep learning models are often considered "black boxes" as they fail to generalize well for novel datasets. Also, this model does not consider the whole protein while building the model so it could be neglecting the useful information. As mentioned in the methods section, the model takes only the designated protein pocket information so as to make predictions. Another limitation of the model is that, the grid box is centered at the 20 Å × 20 Å × 20 Å dimension. All the interactions within the grid box were accounted for. However, long-range interactions such as electrostatic interactions may not be accounted for in the given grid box which does not account for all the interactions between the protein and their respective ligand structures. Also, the features in the grid box of the given dimensions, such as atomic partial charges were computed using the AM1-BCC empirical methods. These may not be giving a correct account of the interactions and may be adding towards a noiser data.

Pafnucy can be used significantly in research revolving around drug design and discovery domains. The impact of small molecule ligands or drugs on target proteins or receptors is a crucial area of research these days. Pafnucy, other than testing a single protein against a ligand can test multiple ligands against a protein as well which makes it all the more useful tool for researchers in expediting the virtual screening and eventually the identification of lead molecules for the drug discovery process.

CHAPTER 3 :: Benchmarking the OnionNet-2 model

3.1. Introduction

3.1.1. A brief about the OnionNet-2 model

As established in the previous sections, virtual screening is a very crucial task in the drug discovery process. It involves accurate prediction of the binding affinity value of the protein and ligand complex. Deep learning methods have significantly improved the accuracy of the virtual screening task due to the ability of DL models to learn from the raw data. In this part of the project, we have benchmarked the DL-based model namely OnionNet-2[49], to predict the protein-ligand binding affinities. It is a Convolutional Neural Network based DL model. The interactions between the PL bimolecular complexes are represented by the number of contacts between the residue atoms in the proteins and the ligand atoms in multiple-distance shells.

3.1.2. Dataset and Feature preparation for the OnionNet-2 model

Dataset Preparation:

The PDBbind dataset version 2019 was used for training purposes. The model was evaluated using the CASF-2016 core dataset and the CASF-2013 core dataset available as a part of the PDBbind dataset. 1000 random PL complexes were taken to validate the model from the refined set. The remaining complexes in the refined set including the general set were used in training.

Feature Generation:

The features used in the model are pair numbers of specific residue(protein) and atom(ligand) contact features in a number of distance shells. The minimum distance between the atoms of the ligand and the residues of the proteins is considered as representative distances. Firstly, each atom in the ligand is taken into consideration and a N number of shells are defined around it. The shell thickness is given as δ, except the first shell which is a sphere with a radius of $d0$. The boundary K_i of the ith shell if given as,

$$0 < \mathbf{K}_i < \ d_0, \quad i = 1$$
$$d_0 + (\mathrm{i} - 2)\,\delta \ \leq \mathbf{K}_i < \ d_0 + (\mathrm{i} - 1)\,\delta, \quad i \geq 2$$

The ligand atoms were classified into 8 atom types, i.e., C, H, N, O, P, S, HAL, and DU. HAL represents the halogen atoms such as F, Cl, Br, and I and DU represents the types of atoms that have been excluded in the previous seven types. The type OTH was given to represent the water, ions and the non-natural residues other than the 20 natural residues.

The residue-atom distance is defined as the distance between the atom in the ligand and the nearest heavy atom in the protein residue. For any shell, the residue-atom contacts have been calculated and used as a feature.

- Each shell has, 8 X 21 = 168 residue-atom contact combinations.
- There are 168 features for a specific shell.
- If N is total number of shells, then N X 168 features will be generated in total.

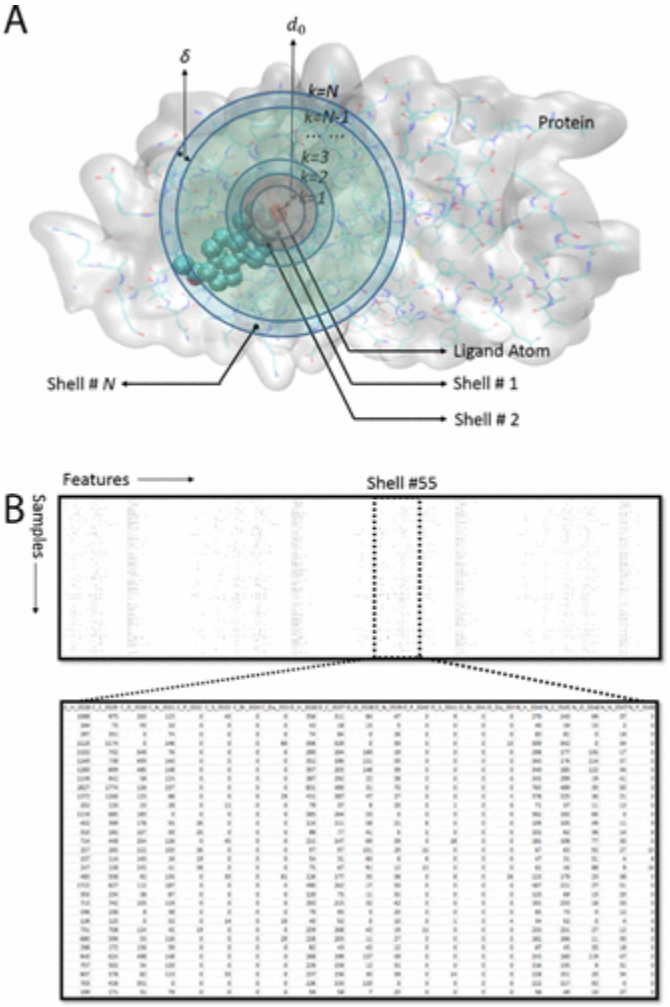

Figure 17: Featurization of protein-ligand complexes based on pair contact numbers of residue-atom features in multiple distance shells[48].

3.1.3. OnionNet-2 Model Architecture and Results

Architecture:

A 2D convolution-based CNN model was used for learning the relation between the contact features and the binding affinity value. The model was constructed using the Keras package in TensorFlow. The raw form of data is pre-processed before feeding into the model. Standardization of features was done using the Sklearn package in Python so as to fit the feature to a Gaussian distribution. The 1D matrices were converted into a 2D matrix to mimic the feature images, which is used as the input for the CNN model. CNNs are composed of various layers and functions. One such layer is the convolutional layer or the ConvLayer. Three convolutional layers were used in the model. The ConvLayer had 32, 64, and 128 filters with size of filter layers as 4 and stride as 1. The result of the last convoluted layer is flattened before passing into the fully connected layer. The fully connected layer integrates the features extracted by the ConvLayers to predict the value of binding affinity, which is represented as p(Kd). Two fully connected layers with 100 and 50 neurons are used before the output layer, which captures the relationship between the characteristic contact-based features and the p(Kd) value. To increase the non-linearity of the model, a ReLU function was added after each ConvLayer and fully connected layer. A batch normalization layer and L2 regularization were used after the fully connected layers.

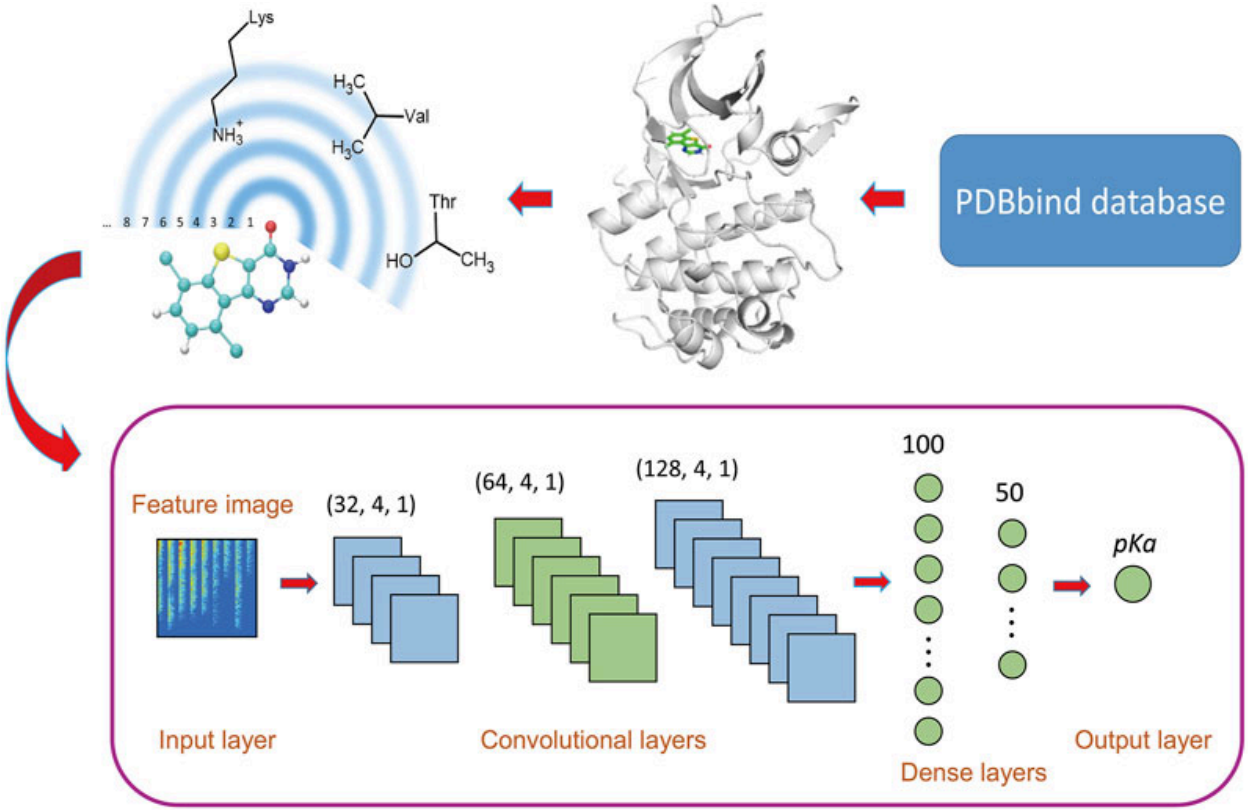

Figure 18: Architecture of the OnionNet-2 model[49].

Metrics:

Dataset	**R**	**RMSE**
Test (v. 2016 core set, PDBbind)	0.864	1.164
Test (v. 2013 core set, PDBbind)	0.821	1.357

Table 4: OnionNet-2 model performance according to paper[49]

The OnionNet-2 model makes use of simple structural features to predict the protein-ligand binding affinity. The model was trained on the true binding PL complexes and not the decoy binder molecules, hence the prediction power can be further improved by including the decoy molecules.

3.2. Materials

3.2.1. Dataset

The refined set of the PDBbind dataset version 2020 was used in the third part of the study. The details of the refined set have been mentioned previously. All the complexes in the refined set were used to carry out the benchmarking task. The model takes the whole protein and the ligand structures as input, so these files were extracted from the refined set. The protein files were in PDB format, while the ligand files were in mol2 or SDF format. The ligand files had to be converted to PDB format.

2.2.2. Files pre-processing: OpenBabel

OpenBabel is a chemical toolbox that allows working with chemical data so as to convert, analyze, search, manipulate, and process data. It is a significant tool used in the field of computational chemistry. It supports many file formats including the most popular ones i.e., PDB, SDF, mol2 etc. It was downloaded and can be used in any directory by suing "obabel" as a prefix in the command. Openbabel was used to covert the SDF files of ligands to PDB format.

3.2.3. OnionNet-2 command-line software

The command line version of the OnionNet-2 was cloned from the GitHub repository. The command line version is developed using Python programming language. Firstly, A designated Python environment named *OnionNet-2* was created and all the required dependencies were installed in the environment. This was followed by using the command further to carry out the prediction tasks. The model predicts the binding affinity in terms of p(Kd).

3.2.4. System Requirements

The repository was downloaded on the department's Linux x86_64 architecture server. Miniconda, which is a small bootstrap version of Anaconda, was installed on the server. It is a free and minimal installer available for conda. It includes conda, Python, and packages that both depend on. For installing additonal packages the command *pip install* is used. Due to the large number of PL complexes to be worked upon, it was rudimentary to use the sever instead of the local machine due to the limited computational power of the local machine.

3.3. Methods

3.3.1. Dataset preparation

The protein and ligand structural files were separated from the refined set of the PDBbind 2020 version. The protein files were in PDB format and ligand files were in mol2 or SDF format. Further, the ligand molecules were converted from SDF to PDB format using the open babel command line toolbox. The commands to carry out the same are mentioned as follows:

The general command to convert all files ending in .xyz extension to PDB files:
obabel *.xyz -opdb -m

In our case, the command would be:
obabel *.SDF -opdb -m

3.3.2. Model Usage

The following steps were followed to benchmark the OnionNet-2 model so as to compute the binding affinity values of PL complexes:

- We need to activate the Python environment, namely, *OnionNet-2,* to use the dependencies that the model requires. The environment is activated using the command:
  conda activate OnionNet-2
- The next step is to navigate to the *scoring* directory where the *predict.py* file is present. The following command is used to score the PL complexes:

  The general command is as follows:
  **python predict.py **
  **-rec_fpath <protein_pdb_path> **
  **-lig_fpath <ligand_pdb_path> **
  **-shape 84,124,1 **
  **-scaler <scaler_file_path> **
  **-model <model_file_path> **
  **-shells 62 **

-out_fpath <outfile_path>

In our case, the command can be modified as follows:
**python predict.py **
**-rec_fpath protein_pdb_path **
**-lig_fpath ligand_pdb_path **
**-shape 84,124,1 **
**-scaler /home/parneet22193/OnionNet-2/models/train_scaler.scaler **
**-model /home/parneet22193/OnionNet-2/models/62shell_saved-model.h5 **
**-shells 62 **
-out_fpath outfile_path

- The output is stored in a CSV file, which gives the complex PDB ID as first column and the p(Kd) predicted value in the second column.
- The above steps can only process one protein-ligand complex at a time so, the task had to be iterated for all the PL complexes in the refined set that had to be tested. This was done using a Python script, which used the *subprocess* module. It wraps codes with Python and allows for simplified and easier use of command-line scripting options. The commands given above could now be run using a Python program. All the PL complexes were iteratively processed using a for loop, and the binding affinity prediction data was appended to a single CSV file.
- Finally, the CSV file had a PDB ID and prediction value columns.

3.3.3. Evaluation of the model results

To compare the actual and predicted values of BA, the units of the same had to be dealt with. The predicted values were in the form of p(Kd), so the actual values of PL binding affinities in the refined set were converted to same scale first. This was done by first converting the values to Molar units and then to respective BA values similar to the predicted values scale. The conversion formula used is :
-log10(IC50 or Ki or Kd) = p(Kd) = Binding affinity
For evaluation of the model, regression metrics such as R-squared value, MAE and RMSE were computed using the sklearn library. This was followed by plotting a scatter diagram for actual vs the predicted values of binding affinity.

3.4. Results

3.4.1. Test result metrics

The model tested on the refined set (PDBbind v. 2020) was evaluated quantatively by computing the following regression metrics.

Test Dataset	R-squared	MAE	MSE	RMSE
PDBbind refined set (v. 2020)	0.854	0.465	0.556	0.745

3.4.2. Test result scatter plot

The scatter diagram showing the plot of actual vs predicted binding affinites is as follows:

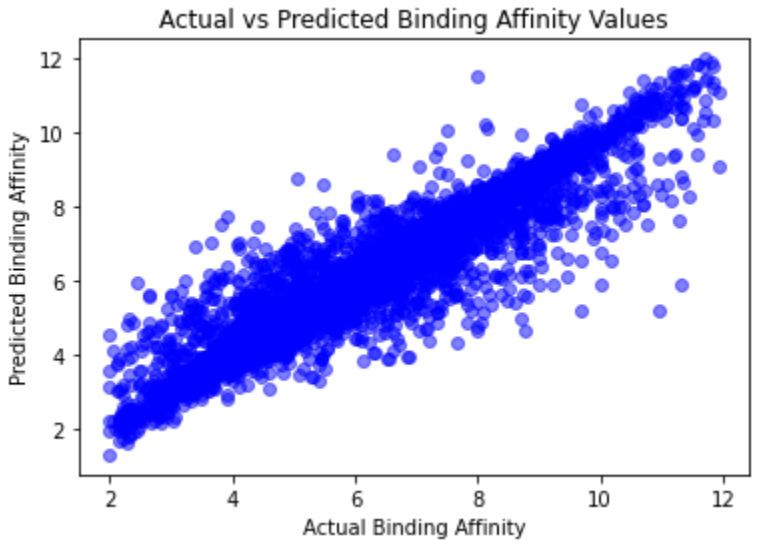

Figure 19: Scatter diagram showing the actual v/s predicted BA using OnionNet-2

3.5. Discussion

To summarize, OnionNet-2 is a two-dimensional CNN-based Deep Learning model that can carry out protein-ligand binding affinity prediction tasks. Hence, predicting the binding affinity values for novel PL complexes can be very useful while undergoing the

virtual screening task in the drug discovery process. The model gave an R-squared value of ~0.854 when we tested it on the refined set of the PDBbind(v. 2020) dataset. This metric is significantly better than the one obtained in our first and second approaches. In the first descriptor-based approach, we obtained an R-squared value of ~0.60, and in the second approach, which involved benchmarking of the Pafnucy model, we obtained an R-squared value of ~0.46.

The OnionNet-2 model performs better than the two other approaches. On comparing the Pafnucy and OnionNet-2 models, it is evident that the Pafnucy model takes into consideration only the pocket information of the protein and the ligand due to which it fails to account for the whole protein molecule while the OnionNet-2 model accounts for the entire protein. This could be one of the reasons why the OnionNet-2 model showed better results on the test set that we used to benchmark both models. Also, the OnionNet-2 model can recognize the physical nature of the ligand-binding pocket interaction.

FUTURE SCOPE

The field of drug dicovery and design is growing very rapidly these days owing to the advancing technologies that are coming up everyday. Our study focussed on predicting the binding affinity of the protein-ligand complexes which is a very crucial task in the drug dicovery process as it allows us to search vast chemical space to extablish lead compunds that can be considered for further research in the process of drug dicovery. The process of ranking a drug molecule against a target protein is a very crucial part of drug discovery. As compared to the experimental techniques employed in the earlier years, the process has been transformed completely into a computational one. This allows for efficiency and less time consumption as compared to the traditional experimental techniques that were used as a part of the virtual screening process. We have developed Machine Learning based approach which relied on the computation of chemical descriptors of protein and ligand molecules to be employed in development of models that can predict the binding affinities. These models were evaluated using regression metrics such as R-squared, mean squared error etc. This method vastly depended on the 1-dimensional feature vector representation of of the PL molecules allowing for scope of improvement. Furthermore, we benchmarked the 2-dimensional and 3-dimensional CNN based models such as Pafnucy and OnionNet-2. These models accounted for spatial features of the protein ligand complexes allowing for the models to be more accurate and accountable to be used in the drug screening purposes.

Although a lot of research has been done in this field of computational chemistry, there is still room for improvement. Predicting the PL binding affinity accurately and efficiently is still an open challenge for researchers due to the selective nature of this task. The process is dependent on a lot of properties of the PL complex such as constitutional makeup, shape, size , physiochemical properties etc. Hence, it is fundamental that the features are selected with utmost precision. The ML models are highle dependent on the feature selection process so as to give accurate results. ON the other hand, DL models can learn from the raw data and the feature selection process is automated by the model architecture itself. Hence, DL will play a significant role in the advancement of the drug dicovery process.

REFERENCES

[1] R. Morris, K. A. Black, and E. J. Stollar, “Uncovering protein function: from classification to complexes,” *Essays Biochem.*, vol. 66, no. 3, pp. 255–285, Aug. 2022, doi: 10.1042/EBC20200108.

[2] Y. Fu, J. Zhao, and Z. Chen, “Insights into the Molecular Mechanisms of Protein-Ligand Interactions by Molecular Docking and Molecular Dynamics Simulation: A Case of Oligopeptide Binding Protein,” *Comput. Math. Methods Med.*, vol. 2018, p. 3502514, Dec. 2018, doi: 10.1155/2018/3502514.

[3] Z. Wang and P. A. Cole, “Catalytic Mechanisms and Regulation of Protein Kinases,” *Methods Enzymol.*, vol. 548, pp. 1–21, 2014, doi: 10.1016/B978-0-12-397918-6.00001-X.

[4] “Kinases and Cancer - PMC.” Accessed: May 12, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5876638/

[5] S. J. Mehdi *et al.*, “Protein Kinases and Parkinson’s Disease,” *Int. J. Mol. Sci.*, vol. 17, no. 9, p. 1585, Sep. 2016, doi: 10.3390/ijms17091585.

[6] G. Klebe, “Protein–Ligand Interactions as the Basis for Drug Action,” in *Drug Design: Methodology, Concepts, and Mode-of-Action*, G. Klebe, Ed., Berlin, Heidelberg: Springer, 2013, pp. 61–88. doi: 10.1007/978-3-642-17907-5_4.

[7] “Protein Binding Pocket Dynamics | Accounts of Chemical Research.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.accounts.5b00516

[8] P. L. Kastritis and A. M. J. J. Bonvin, “On the binding affinity of macromolecular interactions: daring to ask why proteins interact,” *J. R. Soc. Interface*, vol. 10, no. 79, p. 20120835, Feb. 2013, doi: 10.1098/rsif.2012.0835.

[9] N. F. Brás, N. M. F. S. A. Cerqueira, S. F. Sousa, P. A. Fernandes, and M. J. Ramos, “Protein Ligand DockingDockingin Drug DiscoveryDrug Discovery,” in *Protein Modelling*, G. Náray-Szabó, Ed., Cham: Springer International Publishing, 2014, pp. 249–286. doi: 10.1007/978-3-319-09976-7_11.

[10] “Dissociation Constant - an overview | ScienceDirect Topics.” Accessed: May 12, 2024. [Online]. Available: https://www.sciencedirect.com/topics/engineering/dissociation-constant

[11] “Drug Design and Discovery: Principles and Applications - PMC.” Accessed: May 12, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6155886/

[12] “IJMS | Free Full-Text | Molecular Science for Drug Development and Biomedicine.” Accessed: May 12, 2024. [Online]. Available: https://www.mdpi.com/1422-0067/15/11/20072

[13] W. Yu and A. D. MacKerell, “Computer-Aided Drug Design Methods,” *Methods Mol. Biol. Clifton NJ*, vol. 1520, pp. 85–106, 2017, doi: 10.1007/978-1-4939-6634-9_5.

[14] “The Process of Structure-Based Drug Design - ScienceDirect.” Accessed: May 12, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1074552103001947

[15] C. Acharya, A. Coop, J. E. Polli, and A. D. MacKerell, “Recent Advances in

Ligand-Based Drug Design: Relevance and Utility of the Conformationally Sampled Pharmacophore Approach," *Curr. Comput. Aided Drug Des.*, vol. 7, no. 1, pp. 10–22, Mar. 2011.
[16] "Quantitative Structure–Activity Relationship (QSAR) Study Predicts Small-Molecule Binding to RNA Structure | Journal of Medicinal Chemistry." Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.jmedchem.2c00254
[17] "Computational approaches streamlining drug discovery | Nature." Accessed: May 12, 2024. [Online]. Available: https://www.nature.com/articles/s41586-023-05905-z
[18] "ADMET in silico modelling: towards prediction paradise? | Nature Reviews Drug Discovery." Accessed: May 12, 2024. [Online]. Available: https://www.nature.com/articles/nrd1032
[19] G. Sliwoski, S. Kothiwale, J. Meiler, and E. W. Lowe, "Computational Methods in Drug Discovery," *Pharmacol. Rev.*, vol. 66, no. 1, pp. 334–395, Jan. 2014, doi: 10.1124/pr.112.007336.
[20] "Role of computer-aided drug design in modern drug discovery | Archives of Pharmacal Research." Accessed: May 12, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s12272-015-0640-5
[21] R. Meli, G. M. Morris, and P. C. Biggin, "Scoring Functions for Protein-Ligand Binding Affinity Prediction Using Structure-based Deep Learning: A Review," *Front. Bioinforma.*, vol. 2, Jun. 2022, doi: 10.3389/fbinf.2022.885983.
[22] R. Meli, G. M. Morris, and P. C. Biggin, "Scoring Functions for Protein-Ligand Binding Affinity Prediction Using Structure-based Deep Learning: A Review," *Front. Bioinforma.*, vol. 2, Jun. 2022, doi: 10.3389/fbinf.2022.885983.
[23] D. D. Wang, M. Zhu, and H. Yan, "Computationally predicting binding affinity in protein–ligand complexes: free energy-based simulations and machine learning-based scoring functions," *Brief. Bioinform.*, vol. 22, no. 3, p. bbaa107, May 2021, doi: 10.1093/bib/bbaa107.
[24] Z. Guo and R. Yamaguchi, "Machine learning methods for protein-protein binding affinity prediction in protein design," *Front. Bioinforma.*, vol. 2, Dec. 2022, doi: 10.3389/fbinf.2022.1065703.
[25] G. Bitencourt-Ferreira and W. F. de Azevedo, "Machine Learning to Predict Binding Affinity," *Methods Mol. Biol. Clifton NJ*, vol. 2053, pp. 251–273, 2019, doi: 10.1007/978-1-4939-9752-7_16.
[26] R. Wang, X. Fang, Y. Lu, C.-Y. Yang, and S. Wang, "The PDBbind Database: Methodologies and Updates," *J. Med. Chem.*, vol. 48, no. 12, pp. 4111–4119, Jun. 2005, doi: 10.1021/jm048957q.
[27] Z. Liu *et al.*, "PDB-wide collection of binding data: current status of the PDBbind database," *Bioinformatics*, vol. 31, no. 3, pp. 405–412, Feb. 2015, doi: 10.1093/bioinformatics/btu626.
[28] "Binding MOAD, a high-quality protein–ligand database | Nucleic Acids Research | Oxford Academic." Accessed: May 12, 2024. [Online]. Available: https://academic.oup.com/nar/article/36/suppl_1/D674/2507695
[29] "CSAR Data Set Release 2012: Ligands, Affinities, Complexes, and Docking Decoys | Journal of Chemical Information and Modeling." Accessed: May 12, 2024.

[Online]. Available: https://pubs.acs.org/doi/10.1021/ci4000486
[30] “Diverse, High-Quality Test Set for the Validation of Protein−Ligand Docking Performance | Journal of Medicinal Chemistry.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/jm061277y
[31] E. N. Muratov *et al.*, “QSAR without borders,” *Chem. Soc. Rev.*, vol. 49, no. 11, pp. 3525–3564, 2020, doi: 10.1039/D0CS00098A.
[32] Z. Deng, C. Chuaqui, and J. Singh, “Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein−Ligand Binding Interactions,” *J. Med. Chem.*, vol. 47, no. 2, pp. 337–344, Jan. 2004, doi: 10.1021/jm030331x.
[33] “Binding Affinity Prediction with Property-Encoded Shape Distribution Signatures | Journal of Chemical Information and Modeling.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/ci9004139
[34] “Support Vector Regression Scoring of Receptor–Ligand Complexes for Rank-Ordering and Virtual Screening of Chemical Libraries | Journal of Chemical Information and Modeling.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/ci200078f
[35] M. Wójcikowski, P. J. Ballester, and P. Siedlecki, “Performance of machine-learning scoring functions in structure-based virtual screening,” *Sci. Rep.*, vol. 7, p. 46710, Apr. 2017, doi: 10.1038/srep46710.
[36] “SFCscoreRF: A Random Forest-Based Scoring Function for Improved Affinity Prediction of Protein–Ligand Complexes | Journal of Chemical Information and Modeling.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/ci400120b
[37] H. Li *et al.*, “Classical scoring functions for docking are unable to exploit large volumes of structural and interaction data,” *Bioinformatics*, vol. 35, no. 20, pp. 3989–3995, Oct. 2019, doi: 10.1093/bioinformatics/btz183.
[38] “Learning from the ligand: using ligand-based features to improve binding affinity prediction | Bioinformatics | Oxford Academic.” Accessed: May 12, 2024. [Online]. Available: https://academic.oup.com/bioinformatics/article/36/3/758/5554651
[39] “GAUSSIAN PROCESSES FOR MACHINE LEARNING | International Journal of Neural Systems.” Accessed: May 12, 2024. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0129065704001899
[40] H. Wang, “Prediction of protein–ligand binding affinity via deep learning models,” *Brief. Bioinform.*, vol. 25, no. 2, p. bbae081, Mar. 2024, doi: 10.1093/bib/bbae081.
[41] “Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships | Journal of Chemical Information and Modeling.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/ci500747n
[42] “NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein−Ligand Complexes | Journal of Chemical Information and Modeling.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/full/10.1021/ci100244v
[43] “NNScore 2.0: A Neural-Network Receptor–Ligand Scoring Function | Journal of Chemical Information and Modeling.” Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/10.1021/ci2003889
[44] “Review of deep learning: concepts, CNN architectures, challenges, applications,

future directions | Journal of Big Data | Full Text." Accessed: May 12, 2024. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8
[45] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery." arXiv, Oct. 09, 2015. doi: 10.48550/arXiv.1510.02855.
[46] "KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks | Journal of Chemical Information and Modeling." Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.jcim.7b00650
[47] M. M. Stepniewska-Dziubinska, P. Zielenkiewicz, and P. Siedlecki, "Development and evaluation of a deep learning model for protein–ligand binding affinity prediction," *Bioinformatics*, vol. 34, no. 21, pp. 3666–3674, Nov. 2018, doi: 10.1093/bioinformatics/bty374.
[48] L. Zheng, J. Fan, and Y. Mu, "OnionNet: a Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein–Ligand Binding Affinity Prediction," *ACS Omega*, vol. 4, no. 14, pp. 15956–15965, Oct. 2019, doi: 10.1021/acsomega.9b01997.
[49] "Frontiers | OnionNet-2: A Convolutional Neural Network Model for Predicting Protein-Ligand Binding Affinity Based on Residue-Atom Contacting Shells." Accessed: May 12, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fchem.2021.753002/full
[50] "TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions | PLOS Computational Biology." Accessed: May 12, 2024. [Online]. Available: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005690
[51] J. Guo, "Improving structure-based protein-ligand affinity prediction by graph representation learning and ensemble learning," *PLOS ONE*, vol. 19, no. 1, p. e0296676, Jan. 2024, doi: 10.1371/journal.pone.0296676.
[52] "graphDelta: MPNN Scoring Function for the Affinity Prediction of Protein–Ligand Complexes | ACS Omega." Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/10.1021/acsomega.9b04162
[53] "Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities | PLOS ONE." Accessed: May 12, 2024. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0249404
[54] "InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein–Ligand Interaction Predictions | Journal of Medicinal Chemistry." Accessed: May 12, 2024. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/acs.jmedchem.1c01830
[55] "Understanding and interpreting regression analysis | Evidence-Based Nursing." Accessed: May 12, 2024. [Online]. Available: https://ebn.bmj.com/content/24/4/116
[56] "Principle Assumptions of Regression Analysis: Testing, Techniques, and Statistical Reporting of Imperfect Data Sets - Candace Flatt, Ronald L. Jacobs, 2019." Accessed: May 12, 2024. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/1523422319869915

[57] “(PDF) Investigation of performance metrics in regression analysis and machine learning-based prediction models.” Accessed: May 12, 2024. [Online]. Available: https://www.researchgate.net/publication/362719708_Investigation_of_performance_metrics_in_regression_analysis_and_machine_learning-based_prediction_models

[58] R. Guha and E. Willighagen, “A Survey of Quantitative Descriptions of Molecular Structure,” *Curr. Top. Med. Chem.*, vol. 12, no. 18, pp. 1946–1956, 2012.

[59] H. Sun, “A Universal Molecular Descriptor System for Prediction of LogP, LogS, LogBB, and Absorption,” *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 2, pp. 748–757, Mar. 2004, doi: 10.1021/ci030304f.

[60] “Molecular Descriptors and Properties of Organic Molecules | IntechOpen.” Accessed: May 12, 2024. [Online]. Available: https://www.intechopen.com/chapters/58592

[61] “PaDEL‐descriptor: An open source software to calculate molecular descriptors and fingerprints - Yap - 2011 - Journal of Computational Chemistry - Wiley Online Library.” Accessed: May 12, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/jcc.21707

[62] A. Pande *et al.*, “Pfeature: A Tool for Computing Wide Range of Protein Features and Building Prediction Models,” *J. Comput. Biol. J. Comput. Mol. Cell Biol.*, vol. 30, no. 2, pp. 204–222, Feb. 2023, doi: 10.1089/cmb.2022.0241.

[63] “RDKit.” Accessed: May 12, 2024. [Online]. Available: https://www.rdkit.org/

[64] “Mordred: a molecular descriptor calculator | Journal of Cheminformatics | Full Text.” Accessed: May 12, 2024. [Online]. Available: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-018-0258-y

[65] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *Sn Comput. Sci.*, vol. 2, no. 6, p. 420, 2021, doi: 10.1007/s42979-021-00815-1.

[66] “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach - ScienceDirect.” Accessed: May 12, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308019

[67] N. M. O’Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, “Open Babel: An open chemical toolbox,” *J. Cheminformatics*, vol. 3, no. 1, p. 33, Oct. 2011, doi: 10.1186/1758-2946-3-33.