Applied 3D Convolution Neural Network to Predict Hit-to-Lead from Identifying Docking Sites for Hand - Foot - Mouth Disease

Anh Le-Phan
School of Computer Science and Engineering,
International University, VNU-HCM
HCM city, Vietnam
anhphanle271@gmail.com

Phuc-Chau Do
School of Biotechnology,
International University, VNU-HCM
HCM city, Vietnam
dnpchau@hcmiu.edu.vn

Nga Ly-Tu*
School of Computer Science and Engineering,
International University, VNU-HCM
HCM city, Vietnam
ltnga@hcmiu.edu.vn

***Abstract*—Medical drug development has always been one of the major targets in the medical department and due to the complexity of the structure and arrangement of atoms in an amino acid-chain molecule, advanced technological tools, and methods such as Artificial Intelligence (AI) and its applications are relied on and invested to achieve relative goals. Despite the constant improvement in developing a cure, a drug for diseases, the progress in making is difficult and time consuming. For example, the task is taking on dataset to treat Hand – Foot – Mouth Disease (HFMD) as the case is well known and has existed for decades. Our dataset is in Protein Data Bank (PDB) format for every single protein and ligand files that has gone through docking process by using docking tool which we used AutoDock Vina to extract and create the graphs for 3D Convolution Neural Network (3DCNN) to train. This study concentrates on studying a method to find hit-to-lead compounds from identifying docking sites which relies on the binding affinity (BA) and efficacy. To verify them, we visualize it with 3D visualization. Finally, we propose 3DCNN model to predict hit-to-lead compounds and export csv file with only unique complexes containing labels from the HFMD dataset and the prediction.**

***Keywords*— Hit-to-lead, Binding affinity, 3DCNN, Protein-Ligand complexes, Docking Sites.**

I. INTRODUCTION

Despite the constant and ever growing of human education and advancement in technology, humanity is still not prepared for an attack from biological warfare in which Covid-19 has proven the statement. Therefore, drug development has been invested and supplied ever more with multiple techniques and implementations, especially the application of Artificial Intelligence (AI). It provides an efficient approach to handling enormous amounts of data with an abundance of cheminformatics and bioinformatics tools and libraries. Decades of knowledge and experience on experimenting molecules have provided us with the large and trusty dataset to train the machine learning model. The complexity of the molecules challenges many neural networks to alternate themselves to learn except for 3D Convolution Neural Network (3DCNN) [1] as we break down and connect all the compound features into a graph and train them into predicting the binding affinity of protein-ligand complex from docking.

Our dataset provides data with extensive literature review and study on Hand – Foot – Mouth Disease (HFMD) [2] and combining with an assist tool Python Molecular View (PMV) [3] lead to predicting binding affinity (BA) and identifying hit compound from confirming the docking site in 3D environment.

Our contribution of this paper is:

Firstly, we study drug discovery and drug potency to understand how we extract the features of molecules to identify hit-to-lead compounds. We obtain protein-ligand complexes in PDB [13] format and use RDKit tool [4] to distinguish all the significant features of the compounds and then transfer this data over to neural network model to train. There are a few neural network approaches to handle this type of data, we propose 3DCNN [1]. Secondly, provides a compact architecture as it breaks down the data into matrices to search for the pattern in order to accurately predict hit-to-lead. Through this approach, the preferred binding sites and amino acids can be easily discovered with less cost and time.

Finally, we create a Python UI web application with ingrained neural network models and RDKit tool [4] mentioned above to analyze, predict hit-to-lead and display clustering of docking sites and desirable amino acids from newly provided protein-ligand complexes that share the same protein family and 3dmol.js to verify hit-to-lead from 3D visualization.

Following this introduction, related works are presented in Section 2, where the authors will review previous research and briefly review the background related this field. Section 3 will describe our methodology. The results of the experiments are shown in Section 4. Finally, the authors give a conclusion including completed work and suggest future improvements..

II. LITERATURE REVIEW-BACKFROUND

A. Literature review

Other well-known docking tools, AutoDock [5] and AutoDock Vina [6], both are open-source software created by the Olson Lab at the Scripps Research Institute and released in January 2002 and May 2010, respectively, by the Scripps Research Institute. The two in collaboration with PMV [3] to create an easy way see molecules and compounds in 3D. The AutoDock Vina [6] was mainly used to generate protein-ligand complexes for training. It is a global docking based on the geometry of the protein and ligand for fit-to-match results. For each time of running, there are maximum 20 complexes generating. The application of finding the active site in a biological particle requires many replications resulting in many unrelated calculations, which causes difficulty in final analysis, such as manual evaluation and time consuming.

PLIP (Protein-Ligand Interaction Profiler) [7,8] is also unforgettable with going beyond the basic docking calculations to express and present the compound reactions in better clarity. It recognizes several forms of interactions, such as hydrogen bonds, van der Waals contacts, and pi-cation interactions, and provides a complete interaction profile for a particular protein-ligand complex. PLIP is a suitable tool for identifying different interacting amino acids between protein and ligand, which can help in clustering the binding poses. However, the outcome also had to be handled manually which caused a problem in analyzing big data sets.

Finally, SS-GNN which was unveiled in June 2022, acquires many positive traits and techniques of other docking tools by utilizing AI to learn the scoring binding affinity by itself and detect a pattern from all target-drug binding to identify and classify hit compounds [9].

Currently, we consider and propose the 3DCNN model to process the HFMD dataset [2] and potentially identify hit-to-lead compounds.

B. Background

Protein, Ligand and Docking:

Proteins are the building blocks, the gears, and the systems of all organisms on this planet as they work inside the cells and deliver nutrients throughout the body. The foundation of proteins is built upon by hundreds to thousands of smaller units called amino acids, which are made from the most abundant atoms in the world, and they are connected into long chains. These chains can be formed from 20 different types of amino acids and their sequence creates each protein's unique 3-dimensional structure and functionalities. Protein is one of the major types of large biomolecules as it is responsible for catalyzing the biochemical reactions that sustain functionalities of the host. Furthermore, thanks to evolution, proteins have strengthened their ability to bind with other compounds that we can apply drugs to produce desirable results as they bind to receptors which are usually referred to ligands. In protein-ligand binding complexes, the ligand is often than not molecules which bind with its target by docking in protein sites and produce reactions with amino acids.

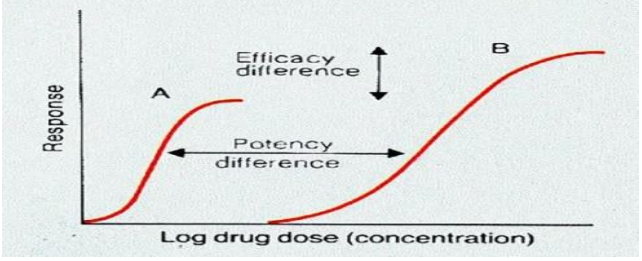

Fig.1 Drug potency [10]

Drug discovery must be followed step by step and its early phase requires finding candidate protein-ligand complexes by applying hit-to-lead method which demands the high drug potency. Potency is a result of the combined score from ligand efficacy and binding affinity. Ligand's ability to manufacture biological response from linking up with amino acids of the target and the quantitative magnitude of the previous response are cited as ligand efficacy (LE). This response may be as an agonist, antagonist, or inverse agonist, depending on the physiological response produced.

The other requirement of potency, BA, measures the tendency or strength of the effect. In general, the higher the affinity the greater attractive forces the better interaction between the ligand and the receptor. This rate can be predicted by utilizing the Docking method which orders the ligand to search for docking sites and rotate itself to create the best results.

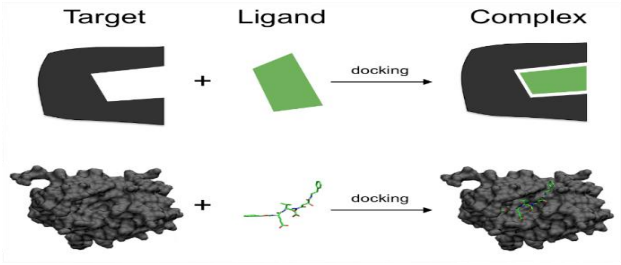

Fig.2 Docking method [11]

Environment and Tools:

Flask [18] is a micro web framework using Python language and supports extensions that append application features for development server, debugger, unit testing and etc. The framework contains Werkzeug which instantiates objects for request, response, utility functions and support Python 2.7 and Python 3.5 and onwards.

3Dmol.js [12] is an object-oriented, webGL based JavaScript library or online molecular visualization without the need to install browser plugins or Java. It provides many features to analyze and visualize the protein-ligand complexes from any format the molecules are in such as pdb, mol2, xyz, sdf and cube. The molecules are displayed in many styles, which also covers the secondary structure design which is the usual requirement.

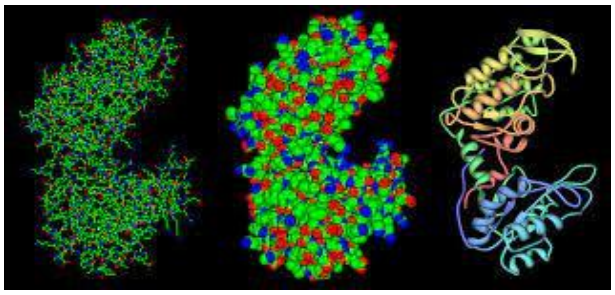

Fig.3 Molecule styles [13]

Rdkit [4] is an open source toolkit which is supported in C++ and Python languages, fortunately for cheminformatics and machine learning software. It scans data in the previously mentioned formats then proceeds to showcase the structure and all features of the molecule. Not only that, we can add draw connections of the atoms in 2D or 3D.

Keras is an open-source Python library that helps understanding and developing Neural Network which has been known since 2005 by Francois Chollet with the help of other libraries like Theano, TensorFlow and CNTK. Keras reduces the workload of developers to integrate their ideas freely while maintaining advanced workflows. With its industry-strength performance and scalability, well-known companies and projects have been utilizing it such as Netflix, Uber, NASA, etc.

3D Convoluational Neural Networks (3D CNNs):

Figure 4 shows 3DCNN is built similarly to normal CNN but with the exception of adding a new dimension within its

shape which is usually preferred to as depth. Everything has the shape of 3 in CNN are now 4 within the layers.

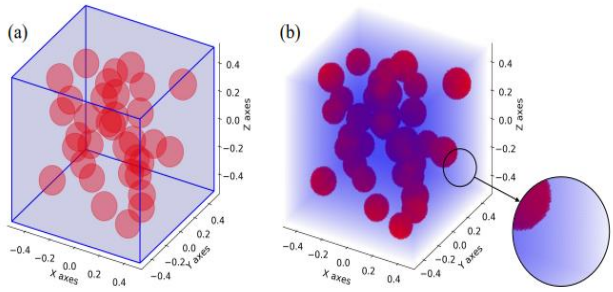

Fig.4 Input data for 3DCNN [1]

Metrics:

For classification, the confusion matrix [15] is brought into play to express the performances smoothly. In addition to the natural usage of Precision and Recall, we have to count Specificity and Negative Predictive Value (NPV) to measure the No hit values as well since those values provide important info on which docking sites we should avoid. Lastly, MCC replaces F1-score to properly find the correlation between Hit and No-hit compounds for Ligands to bind with Protein.

$$Precision = \frac{TP}{PP} \quad (1)$$

Precision is the fraction of relevant instances among the retrieved instances.

$$Recall = \frac{TP}{P} \quad (2)$$

Recall is the fraction of relevant instances that were retrieved.

$$Specificity = \frac{TN}{N} \quad (3)$$

Specificity, True negative rate, is the rate of true negative correctly predicted.

$$NPV = \frac{TN}{PN} \quad (4)$$

NPV displays out the rate of correct negative calls.

The Matthews correlation coefficient (MCC) [16] is a more reliable and dependable metric for taking in the all the four categories of the confusion matrix proportionally to the number of positive and negative elements within the dataset.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}} \quad (5)$$

When the denominator is arbitrarily set to one and any one of the four sums in the denominator is zero, the Matthews correlation coefficient is zero, which can be demonstrated to be the proper limiting value. The MCC, which considers the balancing ratios of the four confusion matrix categories, is more insightful in assessing binary classification difficulties than F1 score and accuracy.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (6)$$

y is the predicted values, x is the original values,n is the total size of the data

MAE is calculated as the sum of absolute errors divided by the sample size: the true value. Alternative formulations may include relative frequencies as weight factors. The mean absolute error uses the same scale as the data being measured.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{\cdot[n\sum x^2 - (\sum x^2)] \cdot [n\sum y^2 - (\sum y^2)]}} \quad (7)$$

Pearson correlation coefficient is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations; thus, it is essentially a normalized measurement of the covariance, such that the result always has a value between −1 and 1.

III. PROPOSAL

Our proposal is shown in Figure 5 which consists of training proposal models, 3D visualization front-end and back-end. The front-end is a web application from using Flask framework with python environment and implementing 3Dmol.js and other tools to visual 3D molecule with Hit-to-lead predictions. The evaluation is completed by utilizing pre-trained proposal models running through preprocessed dataset extracted from RDKit tool [4].

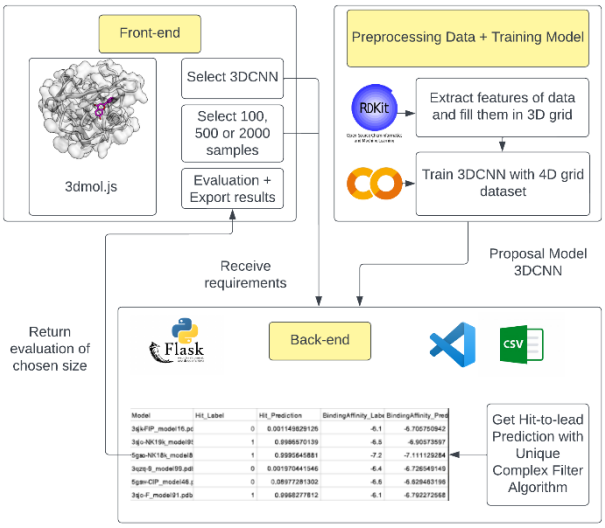

Fig.5. Methodology description

A. Preprocessing

Our dataset is originally taken from RCSB Protein Data Bank (1) and PDBbind (2) which contain thousands of protein-ligand complexes, up to 16000. The raw data goes through its first preprocessing through [2]. By manually analyzing every single molecule from the 16000 datasets using AutoDock Vina and PMV [3] to provide csv files containing the name of the models, the BA score and hit/no hit status of each molecule shown in Figure 6.

	file.pdb	system	model	BA	Hit/No_hit
2	3qzq-10b_model1	3qzq-10b	1	-7.8	hit
3	3qzq-10b_model2	3qzq-10b	2	-7.5	hit
4	3qzq-10b_model3	3qzq-10b	3	-7.5	
5	3qzq-10b_model4	3qzq-10b	4	-7.5	
6	3qzq-10b_model5	3qzq-10b	5	-7.5	
7	3qzq-10b_model6	3qzq-10b	6	-7.5	
8	3qzq-10b_model7	3qzq-10b	7	-7.3	
9	3qzq-10b_model8	3qzq-10b	8	-7.3	
10	3qzq-10b_model9	3qzq-10b	9	-7.2	
11	3qzq-10b_model10	3qzq-10b	10	-7.2	hit
12	3qzq-10b_model11	3qzq-10b	11	-7.2	
13	3qzq-10b_model12	3qzq-10b	12	-7.1	
14	3qzq-10b_model13	3qzq-10b	13	-7.1	
15	3qzq-10b_model14	3qzq-10b	14	-7.1	hit
16	3qzq-10b_model15	3qzq-10b	15	-7.1	hit
17	3qzq-10b_model16	3qzq-10b	16	-7.1	
18	3qzq-10b_model17	3qzq-10b	17	-7.1	
19	3qzq-10b_model18	3qzq-10b	18	-7.1	
20	3qzq-10b_model19	3qzq-10b	19	-7.1	

Fig.6 Original Labels for HFMD

	A	B	C	D	E
1	Model	Hit_Label	Hit_Prediction	BindingAffinity_Labe	BindingAffinity_Pred
2	3sjk-FIP_model16.pc	0	0.001149629126	-6.1	-6.705750942
3	3sjo-NK19k_model9	1	0.9966570139	-6.5	-6.90573597
4	5gao-NK18k_model8	1	0.9995645881	-7.2	-7.111129284
5	3qzq-9_model99.pdl	0	0.001970441546	-6.4	-6.726549149
6	5gav-CIP_model46.p	0	0.08977281302	-6.6	-6.629463196
7	3sjo-F_model91.pdb	1	0.9968277812	-6.1	-6.792272568
8	5gao-FIP_model84.p	1	0.9990828037	-6.7	-6.971171856
9	7dnc-CIP_model38.p	0	8.51E-06	-6.4	-6.001731873
10	5gav-FIP_model4.pc	0	0.01204398274	-6.8	-6.88919735
11	5gav-F_model34.pdl	1	0.9995923208	-6.1	-6.504920006
12	3qzq-FIP_model81.p	0	0.00170292682	-7.4	-6.600903988
13	7dnc-8x_model42.pc	1	0.9998413324	-8.3	-7.791710377
14	5gav-8v_model100.p	1	0.9998098016	-6.7	-7.07869339
15	3sjk-CR_model16.pdl	0	0.0002648073132	-6.2	-7.13373518

Fig.7. 3DCNN Model Prediction for HFMD

Figure 7 shows these two attributes, namely BA and Hit/No_hit, are used for training our proposal neural network model to produce the predictions on the very same attributes. During our validation and testing phase, the models take in the datasets to export the csv files like below, while the remaining attributes are used for matching the data.

B. Hit-to-Lead algorithm

Let us define BA_{org} and BA_{pred} being the scores in 'D' and 'E', respectively, after 3DCNN finished predicting under given dataset. Due to the nature of the original BA computation, there is no definitive way to measure the BA difference. Therefore, we proposal a formula to calsort hit-to-lead compounds from the uncertainty of BA.

$$Error\ BA = |\ BA_{org}\ -\ BA_{pred}| \qquad (8)$$

The model's hit prediction is calculated in the range between 0 and 1, the closer it reaches to 1 and as long as it is above the threshold of 0.9 that the prediction returns a hit status. With the additional assistance from PMV program, we can analyze by monitoring the already best existence docking site then label each complex as 'Hit' or 'Not Hit' with the pdb file to provide the features of the complexes. Not only that, AutoDock Vina is integrated within the program to deliver BA effectively for each ligand model.

C. Build 3DCNN model

The 3DCNN model was originally created to determine the docking sites by identifying hit molecules. Each of its convolutional layers is designed to dissect the grid as if it learns the positions of amino acids and atoms within the grid. The model takes in 1 4D grid data and returns prediction of binding affinity score and the status of hit.

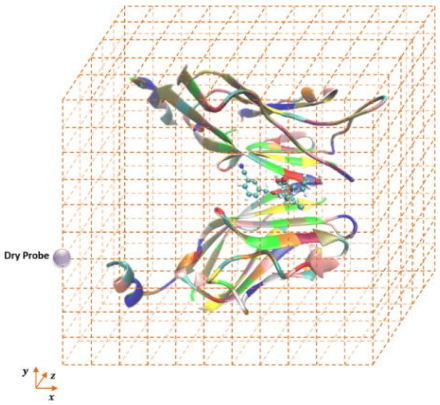

Fig.7. 4D Molecule data [17]

D. Proposed architecture

Figure 8 shows our 3DCNN model constructed upon multiple convolution layers and started out with input layer of (52,52,52,14). Then it repeats the pattern of Conv, Batch Norm, Activation and Pooling until it diverges into Fully connected layers to predict BA and Global pooling to classify hit or no hit.

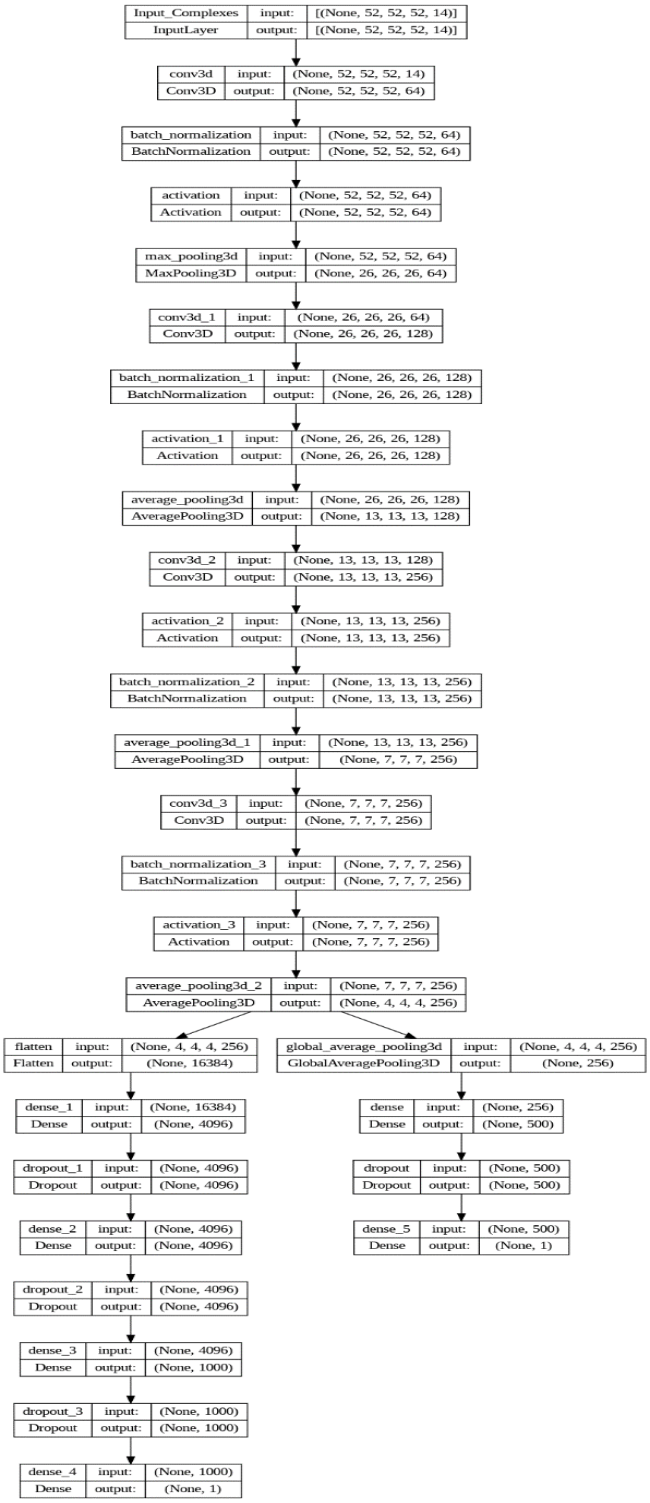

Fig.8 .3DCNN Architecture

E. *Proposed training algorithm*

Let us define N_t is the number of batches for training dataset, N_v is the number of batches for validation dataset, B is the batch size, G_t is the 4D grid data list from training dataset, L_{bt} is the label BA_{org} list of training dataset, L_{ht} is the origin label Hit list of training dataset, G_{vv} is the 4D grid data list from validation dataset, L_{bv} is the label BA_{org} list of validation dataset, L_{hv} is the origin label Hit list of validation dataset, P_b is the BA prediction list, P_h is the Hit prediction list, P_r is the Pearson correlation coefficient score, M_e is the mean absolute error score, M_c is the Matthews correlation coefficient, T_r is temporary score of P_r, T_e is the temporary score of M_e, T_c is temporary score of M_c, P_m is the path to save model, P_{mb} is the path to save best model. Table 1 shows our Pseducode of training 3DCNN model.

TABLE I. PSEUDOCODE OF ALGORITHM 1

Algorithm 1. Proposed 3DCNN training algorithm

No.	3DCNN training algorithm	
1.	Import RDKit, Keras	▷Build model architecture
2.	Choose B =32	▷ To train and omit overload
3.	Assign T_r, T_e, T_c = 0	
4.	Compute N_t using B	▷N_t, training length modulo B
5.	**For** *x* in *range(N_t+1):*	▷ x locate the data batch
6.	Get G_t, L_{bt} , L_{ht} with the length of B	
7.	Model.fit() using G_t, L_{bt} , L_{ht}	▷ To avoid overfitting
8.	Model.save(P_m)	
9.	Compute N_v using B	▷ N_v, validation length modulo B
10.	**For** *y* in *range(N_v+1):*	▷ locate the data batch validation
11.	Get G_v, L_{bv} , L_{hv} with the length of B	
12.	Model.load(P_m)	
13.	Get P_b and P_h from Model.predict() using G_t, L_{bt} , L_{ht}	
14.	Get M_e and P_r from evaluating P_b and L_{bv}	▷ Regression evaluation
15.	Get M_c from evaluating P_h and L_{hv}	▷ Classification evaluation
16.	**End For**	
17.	return M_c,M_e and P_r	
18.	**if** $T_c < M_c$ and $T_e < M_e$ and $T_r < P_r$:	
19.	Assign $T_c = M_c$, $T_e = M_e$ and $T_r = P_r$	
20.	Model.save(P_{mbb})	
21.	**End if**	
22.	**End For**	
23.	return Model.load(P_{mb})	▷ Return the best performed model

IV. RESULT

Our proposal is built with the same building blocks 3D Convolutional layer, Activation layer, Batch Normalization layer, Dropout layer, and Pooling layers which can be accessed by Keras and Tensorflow libraries. These two libraries are written in Python language; thus, our models are trained in Python environment with Jupyter kernel to run each function individually. As the names of 3DCNN suggested, the models work with 3D Convolution which translates to the very 3D Convolutional layer only accepts 4D data grid or 5D data tensor, meaning the data is massive for one molecule meaning Local IDEs would take too long to process the training phase of machine learning. Therefore, we choose Google Colab to take over the entire process from start to end of machine learning. Subscribing to Colab Pro allows us to work with more powerful compute units, we mainly use T4 GPU with High-RAM runtime while withholding 25.5 GB system RAM, and 166.8 GB disk to freely handle machine learning and data analysis. Before starting the machine learning process, the data enters preprocessing phase to match each model's input shape, 3DCNN with (52,52,52,14) and contains all the features of individual atom within the molecule and place them the last dimension.

A. *Dataset*

The original dataset provided [1] contains 8 proteins of the same family having already been bound with ligands. More precisely, each of the proteins only bind with one ligand then get converted into a pdb file for RDKit tool to access.

```
HETATM    1  C   UNL     1       7.052   9.922  10.343  1.00  0.00
HETATM    2  O   UNL     1       7.182   8.733  10.545  1.00  0.00
HETATM    3  N   UNL     1       7.947  10.793  10.851  1.00  0.00
HETATM    4  H   UNL     1       7.844  11.744  10.690  1.00  0.00
HETATM    5  C   UNL     1       5.894  10.426   9.521  1.00  0.00
HETATM    6  C   UNL     1       5.122  11.478  10.318  1.00  0.00
HETATM    7  C   UNL     1       3.799  10.905  10.758  1.00  0.00
HETATM    8  C   UNL     1       2.898  11.697  11.445  1.00  0.00
HETATM    9  C   UNL     1       1.685  11.173  11.850  1.00  0.00
HETATM   10  C   UNL     1       1.372   9.854  11.567  1.00  0.00
ATOM      1  N   GLY A   1      17.860 -17.497   3.252  1.00 34.24
ATOM      2  CA  GLY A   1      17.410 -17.509   1.862  1.00 33.00
ATOM      3  C   GLY A   1      15.882 -17.435   1.717  1.00 30.83
ATOM      4  O   GLY A   1      15.393 -16.455   1.153  1.00 30.60
ATOM      5  H   GLY A   1      18.841 -17.657   3.453  1.00 41.09
ATOM      6  HA2 GLY A   1      17.848 -16.660   1.338  1.00 39.60
ATOM      7  HA3 GLY A   1      17.760 -18.423   1.382  1.00 39.60
ATOM      8  N   PRO A   2      15.100 -18.347   2.334  1.00 30.46
ATOM      9  CA  PRO A   2      13.639 -18.377   2.196  1.00 30.31
ATOM     10  C   PRO A   2      12.899 -17.112   2.659  1.00 28.15
```

Fig.9. Original data of HFMD

The proteins start their structure with ATOM class while the ligands begin with HETATM. Not only that, each ligand traverses on the protein and dock itself one hundred times in order to find the best binding sites, hence there are 100 protein-ligand complexes with the same name. Alongside the pdb files, the author [2] provides a csv file containing crucial info of each molecule, BA and Hit/No_hit, as labels for training our proposed models.

	file.pdb	system	model	BA	LB	UB	Cluster	Hit/No_hit
2	3qzr-10b_model1	3qzr-10b	1	-6.9	0	0	6	
3	3qzr-10b_model2	3qzr-10b	2	-6.6	27.04	32.104	1	hit
4	3qzr-10b_model3	3qzr-10b	3	-6.5	2.516	5.61	6	
5	3qzr-10b_model4	3qzr-10b	4	-6.5	24.517	27.815	1	hit
6	3qzr-10b_model5	3qzr-10b	5	-6.5	27.781	31.051	2	
7	3qzr-10b_model6	3qzr-10b	6	-6.5	25.627	27.47	1	hit
8	3qzr-10b_model7	3qzr-10b	7	-6.5	21.311	24.473	2	
9	3qzr-10b_model8	3qzr-10b	8	-6.4	1.982	5.584	6	
10	3qzr-10b_model9	3qzr-10b	9	-6.4	26.735	31.72	1	hit

Fig. 10. BA and Hit/No hit status of HFMD

Due to the complexity of the model, it completes the testing process within 2 minutes and return astounding results for classification metrics as they are above 97% but only manage an average performance in BA scoring with 44.35% in Pearson coefficient and MAE remains 37.56%. Our model above used the same hyperparameters except for the very structure, Adam optimizer with learning rate of 0.0001 and batch size of 32 while running 100 epochs with Early Stopping under the same Python environment and T4 GPU RAM.

TABLE II. *The* PERFORMANCES *of 3DCNN on 100, 500 and 2000 sample sizes*

Sample Size	Pearson Correlation Coefficient (%)	Precision (%)	Recall (%)	Specificity (%)	NPV (%)	MCC (%)
100	44.357	**100**	**98.148**	**100**	**97.872**	**98.01**
500	44.576	**99.25**	**99.25**	**99.142**	**99.142**	**98.39**
2000	45.798	**98.83**	**98.16**	**98.759**	**98.049**	**96.89**

B. Hit/no Hit/ Hit-to-Lead

The web application operates properly thanks to the Flask framework providing API functions to while allowing access to the local storage. This crucial detail enables other extensions to run within the Python environment, especially RDKit [4] to preprocess the data for 3DCNN [1]. Lastly, based on Unique Complex Filter to sort out the best 10% of the hit-to-lead predictions to distinguish the hit-to-lead compounds so we can observe its docking site with the 3Dmol.js.

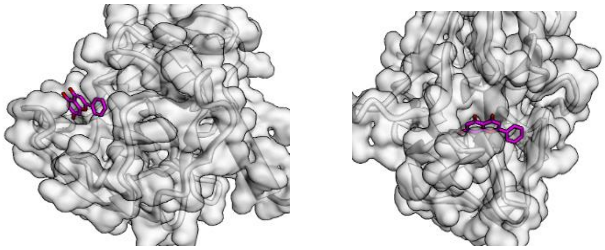

Fig.11. Hit-to-Lead compound – 5c1u-CR_model22

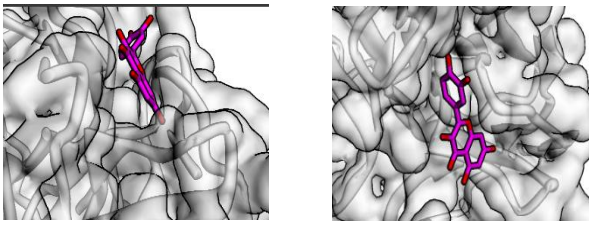

Fig.12. Hit compound – 3qzr-Q_model73

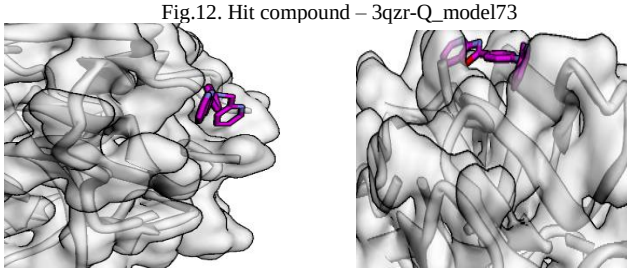

Fig.13. No hit compound – 3qzq-D_model39

TABLE III. *HIT status BY 3DCNN PREDICTION*

5c1u-CR_model22.p	1	0.9996635914	-7	-6.65037488
3qzr-Q_model73.pdb	1	0.9768223166	-5.8	-6.58116340
3qzq-D_model39.pdb	0	0.005138253327	-6.3	-6.47732114

V. CONCULSION

In this paper, we have investigated and understood the drug discovery and drug potency in the making, what are the challenges and the tools we have implemented to solve those challenges. With the help of 3D visualization from 3dmol,js and integrate it into a web application, we can provide means for everyone to understand the subject better. We also propose 3DCNN to handle and assist in searching for hit-to-lead compounds faster. In the future, we will fully implement web applications and enhance the 3DCNN to predict BA better. Not only that we will implement in Graphic Neuron Network (GNN) and other methods to perform hit-to-lead predictions on other proteins in the same family and potentially work on different family of proteins as well.

REFERENCES

[1] 3DCNN Rao, C., & Liu, Y. (2020). Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization. Computational Materials Science, 184, 109850.

[2] Le, T. T. V., & Do, P. C. (2022). Molecular docking study of various Enterovirus—A71 3C protease proteins and their potential inhibitors. Frontiers in Microbiology, 13.

[3] PMV Documentation, Retrieved from: https://www.autopack.org/install/python-molecular-viewer-pmv-installation, last accessed 2023/3/14.

[4] The RDKit Documentation, Retrieved from: https://www.rdkit.org/docs/index.html, last accessed 2023/3/14.

[5] The Scripps Research Institute. (2021). AutoDock. Retrieved from http://autodock.scripps.edu/

[6] Trott, O., & Olson, A. J.: AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of Computational Chemistry, 31(2), 455-461 (2010).

[7] PLIP Homepage. Retrieved from https://projects.biotec.tu-dresden.de/plip-web/plip/index, last accessed 2023/3/14.

[8] Wang, Yu, et al.: Sfcnn: A Novel Scoring Function Based on 3D Convolutional Neural Network for Accurate and Stable Protein–ligand Affinity Prediction. BMC Bioinformatics, vol. 23, no. 1, Springer Science and Business Media LLC, (June 2022).

[9] S. Zhang, Y. Liu, X. Zhang, Y. Chen and J. Wang, "SS-GNN: A Simple-Structured Graph Neural Network for Affinity Prediction," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 6, pp. 2194-2204, (Nov.-Dec. 2020).

[10] WiKiDoc Users. "Ligand (Biochemistry) - Wikidoc." Ligand (Biochemistry) - Wikidoc, 9 Aug. 2012.

[11] Meng, E. C., Shoichet, B. K., & Kuntz, I. D. (1992). Automated docking with grid-based energy evaluation. Journal of computational chemistry, 13(4), 505-524.

[12] 3dmol js Nicholas Rego , David Koes, 3Dmol.js: molecular visualization with WebGL, Bioinformatics, Volume 31, Issue 8, April 2015, Pages 1322–1324, https://doi.org/10.1093/bioinformatics/btu829.

[13] Protein Data Bank, https://www.rcsb.org/, last accessed 30/5/2023

[14] Paciotti, Roberto & Agamennone, Mariangela & Coletti, Cecilia & Storchi, Loriano. (2020). Characterization of PD-L1 binding sites by a combined FMO/GRID-DRY approach. Journal of Computer-Aided Molecular Design. 34. 10.1007/s10822-020-00306-0.

[15] Dipesh Silwal. "Confusion Matrix, Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures". Jan 5,2022.

[16] Huang, S.-Y., and Zou, X. (2010). Advances and challenges in protein-ligand docking. Int. J. Mol. Sci. 11, 3016–3034. doi: 0.3390/ijms11083016.

[17] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., et al. (2000). The protein data bank. Nucleic Acids Res. 28, 235–242. doi: 10.1093/nar/28.1.235.

[18] Flask https://flask.palletsprojects.com/en/2.3.x/, last accessed 30/5/2023.