

# Src kinase app: valid inhibitor generation and prediction with explanation using predictive model and selfies

Mohamed Abd Elaleem<sup>1</sup>

<sup>1</sup>Bachelor of Pharmacy Al-Azhar University - Assiut Branch: Assiut , Egypt

Corresponding author: [mohamed.abdelaleem97@gmail.com](mailto:mohamed.abdelaleem97@gmail.com), [moh.acad@protonmail.com](mailto:moh.acad@protonmail.com)

## Abstract:

Dealing with a small Experimental dataset using a generative model produces a model with underfitting and reduces its ability to generate a new valid compound. Even in the presence of free available chemical databases SMILES string has to use a complex and computationally intensive model to solve validation problems. SELFIES solve all validation problems but further activity optimization is needed with the absence of an app that records molecules generated. In this study, the author uses a predictive model to provide a dataset by a virtual screen of 3 million compounds from a chemical online database in addition to experimental active dataset. Data feed to a different model of one layer Recurrent Neural Network model using both SELFIES and SMILES for about 2-4 epochs. Structure-based drug design was used and Src Kinase as a target to validate both the predictive model and compounds produced by Recurrent Neural Network and further filtration happens using Molecular Dynamics Simulation. SELFIES outperform SMILES in producing valid molecules in all types of Recurrent Neural Network simple structures. Recurrent Neural Network can produce active compounds using the GRU layer without any activity optimization from just 4 runs 100 molecules each. The novelty of the result can be compared to the result coming from predictive model virtual screen data. Recurrent Neural Network can produce novel compounds with key interaction residue with the target protein. All Predictive Models were deployed and ExplainableAI is used to guide generated molecules. MERN stack app SaveMol is used to save molecules produced with substructure research ability and apps links provide here(<https://github.com/phalem/Src>).

## Introduction :

For about the last two decades kinase gain popularity as a drug target[1]. It has an important role in the cell vitality process, differentiation, and survival[2]. It was named due to its ability to catalyze the phosphorylation reaction that the phosphoryl group transferred from ATP (in the presence of Divalent cation such as  $Mg^{2+}$ ) to the protein substrate. Imatinib discovery opens the door to a kinase-selective target [3]. 518 total human genes were identified by Manning et al. including 478 typical and 40 atypical protein kinase genes[2]. Src is a non-receptor protein-tyrosine kinase that associates with oncogenesis the result that came after the Rous sarcoma virus was discovered in 1911[4]. The binding of ASP407 to  $Mg^{2+}$  has a role in the coordinates of the  $\alpha$ - and  $\gamma$ -Phosphates group of ATP. After binding the rest of the loop becomes in an extended conformation and position away from the catalytic center and thus C-terminal(in activation segment) portion provides a place for protein substrate binding. ASP407 is part of DFG[5] that is crucial in the activation of what is called DFG-in and then get out and called DFG-out. Other residues make important interactions like the Salt bridge between Lys298 and Glu313, and some interaction in the hinge residue Glu342 has been reported[6].

Drug discovery is a very long and cost-effective journey with a long feedback loop and a multi-parameter optimization challenge[7]. The process can be accelerated using Artificial Intelligence and its subsets like Machine learning, Deep learning, and Reinforcement learning [8]. It is used in some aspects like QSAR/QSPR and multi-parameter optimization [8], [9].

Supervised machine learning is a type of machine learning that works under supervision on labeled data. Ensemble models were developed to apply machine learning universally like:

(1) Random forest (RF)[10] that takes the average over the  $t$  predictions given [10], [11]  
(2) Support vector machines (SVM) [12], [13] map data into higher dimensions. (3) An artificial neural network (ANN)[14] is a layer of neurons that imitates the way synapses work in the biological brain and work well in classification tasks.

Unsupervised machine learning (which can work on unlabeled data) like clustering was widely used to solve the problem of compound classification[15] without any supervision like Butina Clustering[16]. Butina algorithm can identify homogeneous clusters corresponding to a threshold. It uses SMILES string and encoded as molecular fingerprints, then distance matrix calculated using Tanimoto coefficient within a given threshold. Another commonly used clustering algorithm is K-nearest neighbor (KNN)[15].

The defect of the machine to deal with non-number values lead to convert the structure feature of the molecules to molecular descriptors[17]. There are many widely used descriptors like Molecular ACCess System key (MACCS) which used a binary array to the presence or absence of certain substructure[18] and extended connectivity fingerprints[19] (ECFPs) or a modified version (morgan fingerprint). The later was used to encoding atom-centered radial substructures and the former used to encode predefined substructures respectively[18], [19] and both descriptors are implemented in RDKit[20].

Deep learning is another branch of artificial intelligence that consists of artificial neural networks with several hidden processing layers[14], [21]. Neural Networks is a powerful method that can give us answers that don't rely on molecular descriptor[17] due to its ability to perform extract features from non-feature data and deals with misleading data[22] that can be with low featuring. Deep learning uses some learning functions like stochastic gradient-descent optimization[23] and techniques for hyperparameter optimization like early stopping provided by Deep Learning library like TensorFlow[24]. Graph convolution network[55]: is a type of Neural network that deals with graphs and is used in property prediction[25]–[27]. It used a fixed convolution and an aggregation function that can aggregate information from neighbors. One famous implementation of the Graph convolution network is found in the deepchem library[28], [29].

Exploring chemical space is a very challenging process due to the increased amount of drug-like search space. It is estimate to be about  $10^{60}$  -  $10^{100}$  [30], [31] which could be possibly synthetically accessible[32]. AI can play a key role in exploring a chemical space through a generative model. In the context of drug discovery molecular structure validity, drug-likeness [33] and Synthetic accessibility (SAS) [34] are important parameters that have a computational implementation like what found in RDKit and sascore.py script[35].

SMILES (Simplified Molecular Input Line Entry Systems)[36] string is a molecular representation that has many applications in chemoinformatics and de novo molecular design [37]–[39]. Although the popularity of SMILES string and its usability, it comes with limitations like (1) Any small mistakes in the string (closed and open parenthesis) can give non-valid molecules. (2) String can have more than one structure and thus give more than one molecule structure. The ability to solve these problems can impact the performance of the model and give more attention to learning SMILES string rule itself[40]–[42]. Some work tends to make a new representation[43] like the development of DEEP SMILES by O'Boyle et al. The final trip of modification reaches its goal with the SELFIES molecular representation[44].

Modern deep learning approaches like GAN (Generated Adversarial Networks)[45], [46], Graph Convolution[33], [47] and VAE (Variational AutoEncoder) [48],[49] has been reported. One of the VAE examples that deals directly with the text is Aspuru-Guzik autoencoder[49] from the Aspuru-Guzik group. It takes an input of SMILES and converts it into space and recreates by sampling from the space. Molecule produced isn't valid and increasing training produces molecules with carbon atoms only and complex structure[49] in the opposite of Junction tree VAE has 100% molecule validity from a different group[50].

Recurrent Neural Network is a branch of deep learning with one layer for embedding[51] that deals with text data or sequence [52]. It has the ability to memorize the previous text and learn from the back forward which is a defect in other Deep learning architecture. The use of RNN in De novo drug discovery has taken its place[37] with the ability to generate valid and novel molecules [48], [53]–[55]. The main challenge in this field is the ability to synthesize novel molecules[56]–[58]. It uses SMILES string[36] to produce molecules with some limitations like the need to start learning smiles representation syntax first, and molecular optimization that can be done using Reinforcement Learning [59]–[61]. Other architecture LSTM[53] and GRU[62] was added in recent years due to problems of memorization and vanishing gradient problem. The difference between the two architectures is in using Tanh and sigmoid[57],[67] in a different way and a state gate. LSTM[53] shows better at difficult sequences and GRU[63] is more simple and faster with some nearby results. The use of RNN with selfies[44] can give valid molecule due to RNN ability to generate text and since SELFIES was in a grammar any text can be valid. Other problems of using SELFIES are raised like the ability of RDKit[35] to sanitize the molecules produced and unwanted properties that can give molecules with no drug properties.

RNN has the ability to take a text character by character or an element by element if being in one hot encoding or embedding vectors like  $\mathbf{X} = [\vec{x}_0, \vec{x}_1, \dots, \vec{x}_L]$ . RNN has a layer function which is binary and takes input a character or which in our case a vector of the element  $i$  and takes the output from the previous layer function or  $i - 1$ :

$$f(f \dots f(\vec{x}_0, \vec{0}), \vec{x}_1), \vec{x}_2) \dots \vec{x}_L)(1) \text{ (eq.1)}$$

In RNN a hidden state which call  $h$ , which is all these intermediate outputs from the layer function. It named hidden due to its connection with Markov State.

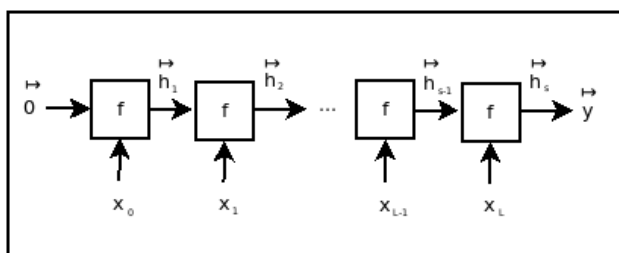


Figure 1: show photo of RNN and hidden state  $\vec{h}$  and the initial hidden state  $\vec{0}$  and the output  $\vec{y}$

The initial hidden state here assuming to be  $\vec{0}$ , with ability to train. Here the same weights and function  $f$  is used. Weights can be reused and thus the choice of parameter number does not depend on the input length, and this also is needed to make RNN accommodate the arbitrary length of input sequences. With a note that length of  $\vec{y}$  may be a function of the input length, so if there is increase in length  $\vec{h}_i$  in each step it will enable increase the length of the output of  $\vec{y}$ .

RNN can also be used in an unsupervised manner like in generative models. Here the task is to try making a prediction of new examples and thus try to learn  $P(\mathbf{X})$ [37]. It can happen through conditioning on a growing sequence, which can predict the sequence of one symbol each time or whats called Autoregressive generation:

$$P(\mathbf{X}) = \prod P(\vec{x}_L | \vec{x}_{L-1}, \vec{x}_{L-2}, \dots, \vec{x}_0)(\vec{x}_1 | \vec{x}_0)P(\vec{x}_0))(1) \text{ (eq.2)}$$

Model output the probability for the next character and a sequence is taken as input. Conditional probability here will be the model by training the network :

$$P(\vec{x}_i | \vec{x}_{L-i}, \vec{x}_{L-i}, \dots, \vec{x}_0)$$

But there are a problem of  $P(\vec{x}_0)$  which solves by determining what the first character is, or making one ourselves as a starting point, so it can be a marker for our starting point.

This training can happen by choosing a split point  $\vec{x}_i$  and trying to train to those proceeding sequence elements (that come from the arbitrary sequence  $\vec{x}$ ), like what happens in multi-class classification, but here characters are the classes and the model give the probability of each class through all classes, and loss is the cross entropy.

Besides all these parameters, it was supposed to use RNN with SELFIES[44]. There is a need to sample from a logit probability or a Temperature to get the first token. This process can give the ability to control the size of a structure that can be produced and thus control some properties like Molecular Weight of the molecules generated. The minimum value of T (Temperature) gives (maximum sampling) and (according to logits sampling) or  $T=1$  and (finally sampling randomly) or  $T = \text{infinity}$ .

High throughput screening is a process to test a very large set of molecules that can reach 100k compounds in the lab. This process increases drug discovery costs [64] with a limited amount of discovery. Computational tools come in handy to test and prioritize compounds library like in Virtual screening that able to screen million to billion of compounds [65]. Tools used like Molecular Docking [66], Protein-ligand interaction and Molecular Dynamics simulation [67]. Molecular Docking is an important step in any drug discovery pipeline that use to determine how active molecules bind to a given protein target through sampling algorithms to understand ligand conformation inside the target binding pocket[68]. Protein-ligand interaction is an important point due to the role of important interactions like the Hydrogen bond that can change the protein binding affinities of the molecule and thus make it difficult for another compound to interact with the target[69]. PLIP[70] is a famous open source tool that gives a protein interaction insight and residue involved in the interaction with the python library[71] included. Molecular Dynamics simulation[67] is one of the simulations that simulate the protein-ligand interaction according to the newton law of motion.

Parameterization like Force field describes what atomic forces contribute what govern and happen in Molecular Dynamics like in GROMOS[72], CHARMM[73], AMBER[74], [75]. Libraries provide a forcefield like OpenMM Forcefields[76]. Molecular Dynamics(MD) gives a piece of detailed and accurate information about what happens inside the system and the reason behind the compound activity and augments the result of molecular docking[68].

OpenMM[77] provides MD simulation with GPU support and a python module that makes it easy to perform a molecular dynamics simulation with less time and cheap results.

Due to the black-box character of the machine learning model and the unknown reason behind taking prediction decisions [78], Explainable Ai[79] comes in handy to provide an interpretation of the decision-making of the model[79], [80]. Many methodologies involve ExplainableAI like uncertainty estimation[80] and feature attribution[81] and instance-based molecular counterfactual explanation has been reported[82].

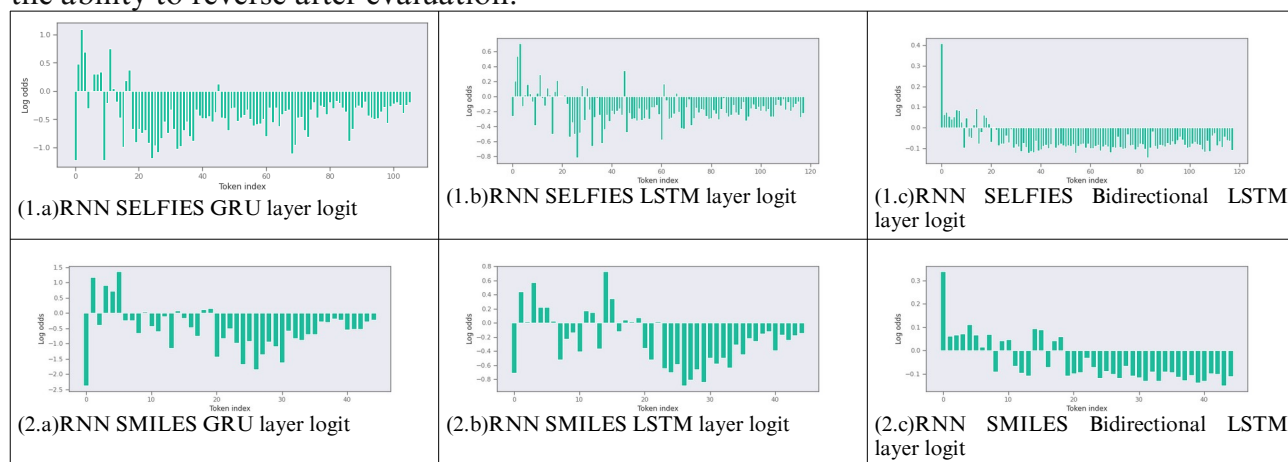
MERN stack: is using technologies based on Javascript[83] to make a full stack website. It uses ReactJS framework[84] to make a Front-end part and ExpressJS framework[85] for the backend part with NoSQL database MongoDB[86] non-relational database. It can produce easy and scalable applications and make it the choice of dealing with a scalable application that is fast and reliable. In this study, the author combined all modern AI techniques to get data that can be fed into a simple RNN model. A comparison between using SELFIES and SMILES was performed in a simple and less computational environment. RNN with SELFIES produced valid compounds and in the case of GRU produce a valid and active result. Models deployed as a real-time web app give the user ability to discover, virtual screen and generate molecules using this approach. ExplainableAI(Explain) part in the app can guide the user by explaining the prediction provided by the model. Generate and Explain can generate molecules and give user insight into further modification or explanation of why compounds generated were inactive. The author provides a place to store molecules that were generated and retrieve it anytime with substructure searching. Ability to produce an active compound that can inhibit target protein with less computational effort open the door to replicating the process for all kinase. Provides an easy road to Reinforcement Learning without any concern about a valid part and the user's sight to optimize compound according to that explanation. The work makes a new way of model validation ability by exploring which residues can be reached by each model. This is an initial step to the discovery of a novel drug for Src kinase and further investigation of Synthetic accessibility needs with the shortage of using a predictive synthetic model with no cost. Exploring the chemical space of the drug-

like molecules can be an easy task with the hope to make a KLIFSAI that makes the same iterative approach to all protein kinases.

## Result and Discussion:

### 1. Recurrent Neural Network:

1.1. Model evaluation: figure(2) show the logit probability of the different models including GRU, LSTM(4 epochs training for both), Bidirectional LSTM(2 epochs training). Same model used for both SELFIES and SMILES string model(All mode have an embedding layer and RNN layer). The model used for further investigation is model (1. a) which uses a layer of GRU and SELFIES as a language. GRU SELFIES was the only model that produce drug-like molecules. SMILES token is 45 due to the exchange of 2 characters atoms with one with the ability to reverse after evaluation.



Figure(2) show logit odds of different simple one layer RNN (a,b,c) using both SELFIES(1) and SMILES(2), Novel compounds comes from the layer of figure (1.a) GRU layer that used SELFIES

1.2. Model scoring: table(1.a,b) show each distribution for different models structure and corresponding Temperature used, with no details about SMILES as no valid molecules were generated. SMILES models found in supplementary. Drug QED mean, drug score range, SAS score, docking range and protein interaction were used to evaluate model generation.

The result shows that in the case of using GRU model compound generated is valid and produces novel molecules. Other structure of RNN produce only valid compound with a very long chain carbon or doesn't obey QED rule and consist of compounds that have atoms rather than carbon like S, N, I and Br. Models and compounds produced are found in the supplementary for further activity investigation and reproducibility. A model like LSTM comes with just 4-5 compounds of small molecules(4-5 from 400 compounds), with some temperature tuning. The amount of training is 4 epochs for (LSTM, GRU) and 2 epochs for Bidirectional LSTM. In SMILES string all model with the same structure and the same amount of training doesn't produce valid compound due to syntax error of SMILES produced and less computational effort. Study doesn't consider other complex and computationally intensive result, the study investigates a way that can be done on a normal computer or on free cloud storage provided as the process can be produced easily for another target or reproduced without a lack of cost. A model with GRU deployed on the web app and other models and SMILES produced will be provided in the supplementary material with a notebook to load and reuse to check the result.

In the case of GRU that uses SELFIES most compound produced from RNN needs to have a multi-objective optimization, no molecules show an outlier, and most compounds are under 1000 molecular weight. The multi-objective approach will make it better, however temperature tuning can produce a compound with less molecular weight and size. Some molecules give high docking score but it appears to have two molecules together like the outlier that appears in the table(-13.4 score). The only way to recognize molecules like that is by viewing in NGL view as it hasn't a valid structure in the NGL. In the context of RNN

temperature number when increase temperature a simple molecule is produced and decreased it gives a very complex and big structure.

Table(1.a): Evaluation of RNN models using SELFIES as a representation. Compounds generated resulting from 100 molecule generation for each model temperature. Count indicate a valid compound produced and QED and SAS range value. Docking and protein interaction result on protein(PDB\_ID:7NG7) pocket center of (-17.300,-2.055,-5.938) and pocket size of (28.0,28.0,28.0).

Temperature	Layer type	Source	Valid Count	Docking affinity range[kcal/mol]	Docking affinity mean[kcal/mol]	QED range	QED mean	SAS range
RNN T=0.15	GRU	SELFEIS	99	-1.0 - -12.8	-7.89	0.02-0.82	0.36	1-7.32
RNN T=0.25	GRU	SELFEIS	96	-1.0 - -13.4	-7.34	0.04-0.87	0.42	1.05-7.36
RNN T=0.5	GRU	SELFEIS	82	-0.7 - -12	-6.8	0.07-0.88	0.4	1.49-8.47
RNN T=0.75	GRU	SELFEIS	85	-2.0 - -10.7	-6.87	0.03-0.76	0.38	1.5-8.4
RNN T=0.15	LSTM	SELFEIS	100	None	None	0.02-0.54	0.13	1-7.8
RNN T=0.25	LSTM	SELFEIS	99	None	None	0.01-0.54	0.16	1.5-8.4
RNN T=0.5	LSTM	SELFEIS	99	None	None	0.00-0.55	0.19	1.6-8.4
RNN T=0.75	LSTM	SELFEIS	99	None	None	0.00-0.67	0.21	2.2-8.4
RNN T=0.15	Bidirectional LSTM	SELFEIS	99	None	None	0.02-0.5	0.22	3.4-8.4
RNN T=0.25	Bidirectional LSTM	SELFEIS	97	None	None	0.01-0.46	0.19	3.6-8.4
RNN T=0.5	Bidirectional LSTM	SELFEIS	100	None	None	0.00-0.54	0.20	4-8.4
RNN T=0.75	Bidirectional LSTM	SELFEIS	99	None	None	0.01-0.56	0.24	3.7-8.4

Table(1.b): show protein ligand for RNN GRU SELFIES model using PLIP with same pocket and protein id of table (1.a), count of each temperature found in table(1.a).

Temperature	H_bond range	H_bond mean	hydrophobic range	hydrophobic mean	SALT_BRIDGE range	PI_STACKIG range	PI_CATION range	HALOGEN range
RNN GRU T=0.15	0-9	2	0-18	8	0-3	0-1	0-2	0
RNN GRU T=0.25	0-7	2	0-19	6	0-2	0-1	0-2	0-1
RNN GRU T=0.5	0-7	2	0-16	6	0-2	0-1	0-2	0-1
RNN GRU T=0.75	0-9	3	0-14	5	0-4	0-1	0-2	0-2

## 2-Predictive model:

2.1.Data preparation: the author finalized 8753 (3253 active compounds and 5500 inactive compounds) and 4919 compounds with a pIC50 data point. The data was big but some compounds are not included in the data because it is unspecific and lack of information about their mode of action and concentration.

### 2.2.Model Evaluation:

2.2.1.Machine learning: Figures (3. a,b) shows the AUC(Area under the ROC Curve) of using MACCS key and morgan fingerprint respectively. Figure(3.c,d) shows evaluation of Random forest regression for pIC50 predicted versus actual value using both MACCS key and morgan fingerprint respectively. From data and figures and table(2.a,b) the author concludes that:

1. Using maccs or morgan fingerprint is not enough in the case of feature numbering, however using a descriptor like modellar and Padel that can provide a large number of features was time-consuming and affect model deployment and the rate of model prediction per molecule.
2. Using a combination of models that use the same fingerprint(like SVM, ANN, RF or RF\_2, ANN\_2) gives an Excellent prediction and increases the ability of the model to get the active compound accurately. There is a lack of molecules that have a high prediction of activity according to all models and thus making it difficult to provide evidence for all model and model combinations (please, see HOW\_TO\_VS notebooks in the supplementary). The problem of understanding the result that comes from the model combination can be solved with a virtual screen of big amount of

data and evaluated using a structure-based approach and thus can provide evidence for all model ability, however it has a computational cost.

2.2.2. Neural Networks: figure (3. e,f) shows the predicted versus actual value plot of using MACCS key and Morgan fingerprint respectively. From data and figures and table(2.a,b) the author concludes that Neural Networks have a great deal in the case of data, but it is not compared when using a combination(RF, SVM, ANN) together or using GCN.

2.2.3. Graph Convolution Network: Figures (3. g,h) show classification model evaluation via accuracy metrics using SMILES and canonical SMILES respectively. (3. i,j) show predicted versus actual using GCN regression in both SMILES and canonical SMILES respectively. Canonical smiles show evidence of increasing prediction accuracy. Using SMILES in regression tasks makes a good prediction than canonical SMILES. The author doesn't know why this might happen is that due to randomness or due to a different amount of training in each case( number of epochs is the same in each case). Regardless, Canonical SMILES as input show a better result than the use of SMILES alone.

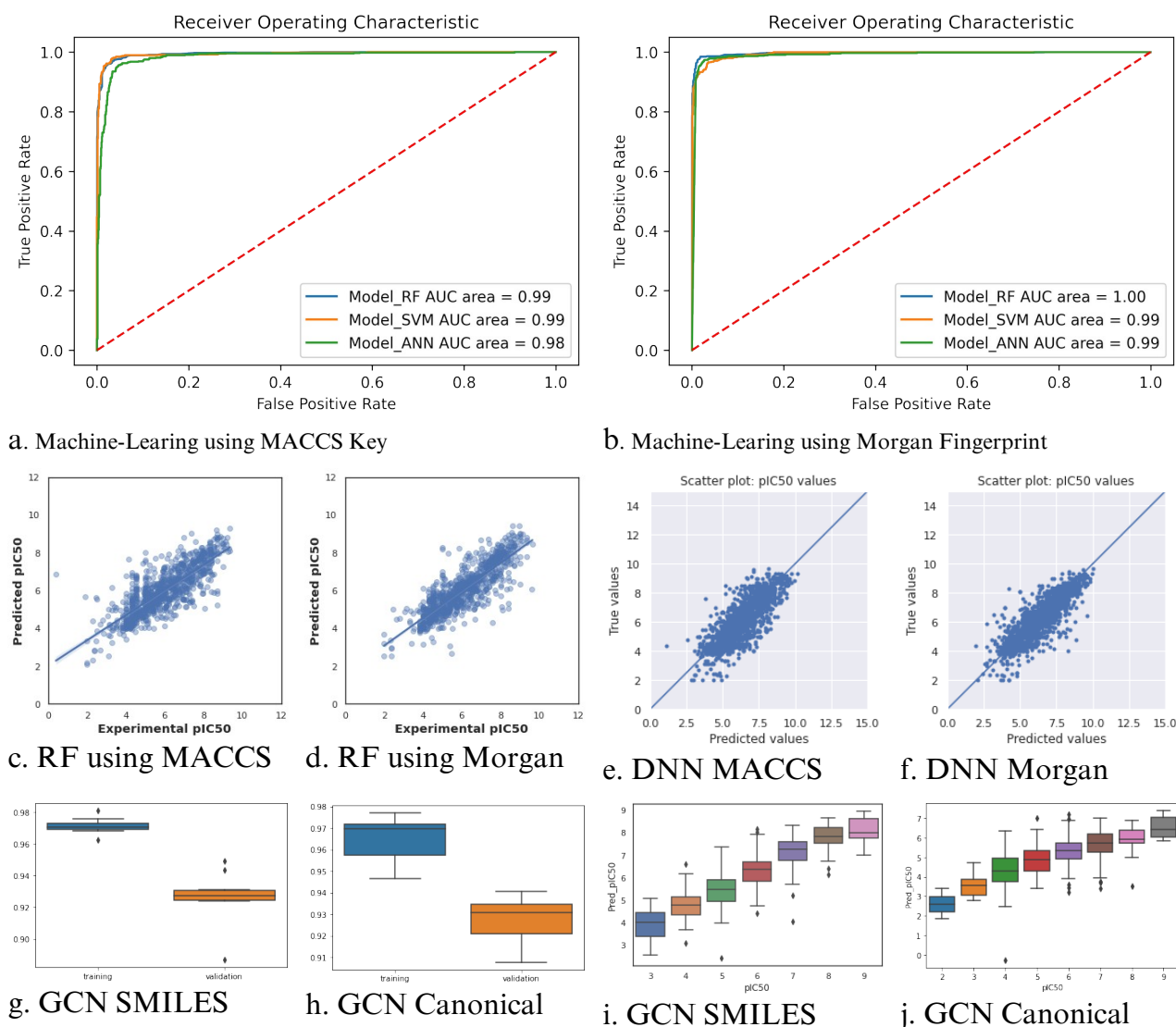


Figure 3: **Show predictive model on bioactivity data** (a,b) show Machine learning classification task evaluation based on MACCS Key, Morgan Fingerprint respectively. (c,d) Random Forrest predictive pIC50 regression task based on MACCS Key, Morgan Fingerprint respectively.(e,f) Deep Neural Network(DNN) model pIC50 prediction model based on MACCS Key, Morgan Fingerprint respectively.(g,h) Graph Neural Network classification based on SMILES and Canonical SMILES respectively via accuracy as evaluation parameter. (i,j) Graph Neural Network pIC50 regression predicted versus actual pIC50 based on SMILES and Canonical SMILES respectively

### 2.3. Model validation and scoring:

2.3.1. Prediction: There are 5788 compounds predicted as active according to all models. Clustering was performed and redundant data was removed after clustering. Compounds were

reduced to 3512 compounds. Due to the time-consuming and resource of docking and protein-ligand interaction a virtual screening of a large database of 2.7 million compound make it difficult to happen. Providing a small dataset gives an accurate prediction of each model capability and residues reached by each model.

2.3.2. Scoring: table(2) show every model name according to Src kinase app and model docking range and residues reached by molecular docking according to PDB id (7NG7). Detailed protein-ligand interaction found in Github or with supplementary. The user can determine which model shows the best model prediction and ability to get the active compound by looking at validation and residues reached by each model( a guide for the user). Using a combination of models is better. Here, the author can't provide details about using a combination scenario because of the limited amount of data which is 39k only. The author provides a Jupyter notebook of HOW\_TWO\_VS and provides a way to make a virtual screening for your data that comes from Src app and also very detailed information about every model and each residues reached in model\_validation, Predictive in Github or in supplementary. Molecules can be found active according to different models, so total count is more than total amount mentioned above.

Table(2.a): show every model produced using a model name in the app and virtual screening result of using it on 50k ZINC data(reduced to less than 5k). Result based on docking and protein interaction on protein(PDB\_ID:7NG7) pocket center of(-17.300,-2.055,-5.938) and pocket size of(28.0,28.0,28.0) same as table(1.a).

Model Name	Model type	Task	Threshold	Fingerprint	Count	Docking affinity range [kcal/mol]	docking affinity mean [kcal/mol]
RF_is_active_1	Random Forrest	Classification	-	Maccs Key	520	-2.3 - -11.2	-9
SVM_isactive_1	Support Vector Machine	Classification	-	Maccs Key	34	-3.9 - -11.1	-8.94
ANN_isactive_1	Artificial Neural Network	Classification	-	Maccs Key	988	-2 - -11.6	-9
RF_isactive_2	Random Forrest	Classification	-	Morgan Fingerprint	109	-7.3 - 11.7	-9
ANN_isactive_2	Artificial Neural Network	Classification	-	Morgan Fingerprint	937	-2 - -11.2	-8.9
RF_pic50_1	Random Forrest	Regression	7	Maccs Key	329	-5.9 - -11.1	-9
RF_pic50_2	Random Forrest	Regression	7	Morgan Fingerprint	34	-3.6 - -11.1	-8.9
NN_predicted_pIC50	Deep Neural Network	Regression	7	Maccs Key	502	-2 - -11.3	-8.96
NN_predicted_pIC50_2	Deep Neural Network	Regression	7	Morgan Fingerprint	130	-6.7 - -10.9	-8.88
GCN_Positive	Graph convolutional network	Classification	0.5	Smiles string	86	-7.5 - -11.1	-9
dch_pic50	Graph convolutional network	Regression	7	Smiles string	62	-5.7 - -10.4	-8.72
GCN_Positive_2	Graph convolutional network	Classification	0.5	Canonical Smiles	1163	-3 - 11.8	-9
dch_pic50_2	Graph convolutional network	Regression	7	Canonical Smiles	17	-7.6 - -10.5	-8.94

Table(2.b) continue from table(2.a) with protein interaction on same protein and pocket of describe on table(2.a) with figure id describe each model.

Model Name	figure	H_bond range	H_bond mean	hydrophobic range	hydrophobic mean	SALT_BRIDGE	PI_STAC KING	PI_CATI ON range	HALOGEN range
------------	--------	--------------	-------------	-------------------	------------------	-------------	--------------	------------------	---------------



						range	range		
RF_isactive_1	4.a	0-10	3	0-14	6	0-3	0-2	0-2	0-2
SVM_isactive_1	4.a	0-7	3	1-12	6	0-2	0-1	0-1	0
ANN_isactive_1	4.a	0-10	3	0-14	6	0-3	0-2	0-3	0-2
RF_isactive_2	4.b	0-8	3	0-12	6	0-3	0-1	0-2	0-2
ANN_isactive_2	4.b	0-9	3	0-15	6	0-3	0-1	0-3	0-2
RF_pic50_1	4.c	0-9	3	0-12	6	0-4	0-2	0-2	0-2
RF_pic50_2	4.d	0-7	3	1-12	6	0-2	0-1	0-1	0
NN_predicted_p IC50	4.e	0-10	3	0-14	6	0-3	0-2	0-3	0-2
NN_predicted_p IC50_2	4.f	0-8	3	0-12	5	0-4	0-2	0-2	0-1
GCN_Positive	4.g	0-7	3	2-13	6	0-2	0-1	0-2	0-1
dch_pic50	4.i	1-8	3	0-10	5	0-2	0-1	0-3	0-1
GCN_Positive_2	4.h	0-11	3	0-14	6	0-4	0-2	0-3	0-2
dch_pic50_2	4.j	0-7	2	1-9	6	0-2	0-1	0-1	0-1

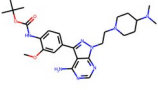
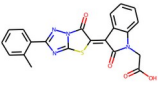
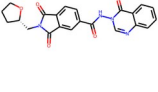
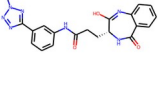
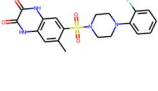
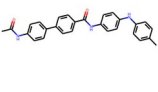
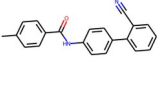
2.3.3. Novels: table(3) shows highly active compounds and docking values, hydrogen bond numbers for both the Predictive bioactivity model and RNN model. Detailed interaction found in GitHub and supplementary.

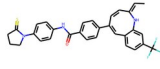
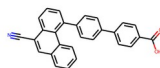
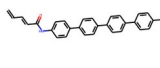
Table(3): Describe Novel compounds docking and protein ligand interaction on the same protein and pocket of table 2 and RNN novel compound T indicate Temperature and V indicate virtual screen process number. Novel structure will appear in next table(Table(4))

name	Produced by	Docking value	QED	hydrogen bond	other interactions
UCW	ZINC Database	-11.4	0.49	344 MET A,341 THR A,407 ASP A,407 ASP A,298 LYS A,342 GLU A	PI_CATION: 298 LYS A,
ZINC00000 2391833	ZINC Database	-10.7	0.54	298 LYS A,407 ASP A,341 THR A,348 SER A,351 ASP A,341 THR A,	None
ZINC0000 17848357	ZINC Database	-11.6	0.64	298 LYS A,341 THR A,407 ASP A,407 ASP A,	None
ZINC0000 95478729	ZINC Database	-10.1	0.59	298 LYS A,341 THR A,280 CYS A,281 PHE A,394 ASN A,341 THR A,389 ASP A,393 ALA A,407 ASP A,	SALT_BRIDGE: 407 ASP A, PI_CATION: 298 LYS A,
ZINC00000 6900314	ZINC Database	-11.1	0.62	341 THR A,348 SER A,298 LYS A,394 ASN A,407 ASP A,	SALT_BRIDGE: 407 ASP A, PI_STACKING: 408 PHE A,
T025V12	RNN T=0.25	-11.3	0.32	341 THR A,407 ASP A,407 ASP A,	PI_CATION: 298 LYS A,
T05V56	RNN T =0.5	-10.7	0.76	344 MET A,407 ASP A,	None
T05V52	RNN T =0.5	-12	0.32	298 LYS A,351 ASP A,	None
T015V10	RNN T =0.15	-12.1	0.34	None	PI_CATION: 298 LYS A,
T015V5	RNN T =0.15	-11.4	0.25	407 ASP A,407 ASP A,	PI_CATION: 298 LYS A,

2.3.4. Molecular Dynamics: After scoring with molecular docking the result from both RNN and predictive model was evaluated via Molecular Dynamics as an accurate metric to inform the result and support molecular docking and protein-ligand interaction. Table(4) show each compound and its interaction energy according to reference ligand(UCW).

Table(4): Continue from Table 3, here the result of Molecular Dynamic simulation of novels compound and filtered compound with time indicate in nanosecond and protein-ligand interaction that occur in % of times in Molecular Dynamics Simulation. Interaction energy is compared to the original ligand in order to evaluate how potent the compound. Detailed interaction information found in the supplementary.

Structure	Publish_name	Molecular Dynamics time	Interaction energy	Protein-ligand interaction occur 30%	Protein-ligand interaction occur 50%	Protein-ligand interaction occur 90%
	UCW	20n	-83.13 ± 5.38 kcal/mol	H_bond: ASP407.X, GLU342.X, PiStacking: TYR343.X, PiCation: LYS298.X, Hydrophobic number:24	H_bond: ASP407.X, GLU342.X, PiStacking: TYR343.X, PiCation: LYS298.X, Hydrophobic number:23	H_bond: ASP407.X, GLU342.X, Hydrophobic number:19
	ZINC 2391833	20n	-62.51 ± 4.99 kcal/mol	H_bond:SER348.X, THR341.X, PiStacking: TYR343.X, PHE408.X, Hydrophobic number:19	H_bond: THR341.X, PiStacking: TYR343.X, PHE408.X, Hydrophobic number:16	H_bond: THR341.X, PiStacking: PHE408.X, Hydrophobic number:12
	ZINC 17848357	20n	-65.39 ± 5.10 kcal/mol	H_bond: THR341.X, LYS298.X, PiStacking: PHE408.X, Hydrophobic number:19	H_bond: LYS298.X, PiStacking: PHE408.X, Hydrophobic number:18	PiStacking: PHE408.X, Hydrophobic number:18
	ZINC 95478729	20n	-66.78 ± 5.20 kcal/mol	H_bond: LYS298.X, Hydrophobic number:18	H_bond: LYS298.X, Hydrophobic number:17	Hydrophobic number:8
	ZINC 6900314	20n	-62.43 ± 5.67 kcal/mol	PiStacking: PHE408.X, Hydrophobic number:17	PiStacking: PHE408.X, Hydrophobic number:15	Hydrophobic number:11
	T025V12	20n	-62.43 ± 3.44 kcal/mol	H_bond: ASP407.X, PiStacking: TYR343.X, Hydrophobic number:19	H_bond: ASP407.X, PiStacking: TYR343.X, Hydrophobic number:18	H_bond: ASP407.X, Hydrophobic number:13
	T05V56	20n	-57.28 ± 3.62 kcal/mol	H_bond: ASP407.X, THR341.X, PiStacking: PHE308.X, Hydrophobic number:16	H_bond:ASP407.X, THR341.X,Hydrophobic number:15	H_bond: ASP407.X, Hydrophobic number:12

	T05V52	5n	-65.59 ± 5.35 kcal/mol	Hydrophobic number:19	Hydrophobic number:18	Hydrophobic number:14
	T015V10	5n	-56.91 ± 3.56 kcal/mol	H_bond: PHE408.X, SER348.X, PiStacking: PHE408.X, Hydrophobic number:18	H_bond: PHE408.X, PiStacking: PHE408.X, Hydrophobic number:17	Hydrophobic number:11
	T015V5	5n	-59.73 ± 3.00 kcal/mol	H_bond: ASP148.X, ASP148.X, Hydrophobic number:19	H_bond: ASP148.X, Hydrophobic number:18	Hydrophobic number:11

The ability to get a ligand potent like Experiment ligand is very hard. Compounds show activity other than it and protein-ligand interaction more than the ligand, but a ligand still has the priority due to its Experimental result. Some compounds show activity better in the docking process, but doesn't enter MD simulation due to some error in molecular dynamics. Error raised due to the workflow the author used and RDKit ability of curated compounds. Most ligands in 5 nanoseconds show decrease in protein-ligand interaction and thus give us an overview of how weak the result when using docking, nevertheless some interaction happens in both.

2.3.6. Molecular Dynamics Renumbering: After making MD simulation the author deals with the problem of renumbering as all residue in the right place x,y,z, but the only number resets from 260 to 1. To solve the problem the author makes a script to align atom with same dimension and place and same residue together and put our reference complex PDB with the complex PDB that workflow makes before simulation and renumbering residue manually in the notebooks. The way author renumber is found in the GitHub or in supplementary. Reference ligand was provided as an example with the same for all novels, please take a look at it in the GitHub repo renumbering problem file and or in supplementary. Besides all these challenges and time-consuming and computational costs the author can't repeat the process, but for anyone who will use the same protein id, it starts with the number 260 and it is very important to notice. Author notebook makes renaming very efficient and numbering residue exactly as it is and an example found in GitHub with reference ligand.

## 2.4. Novels:

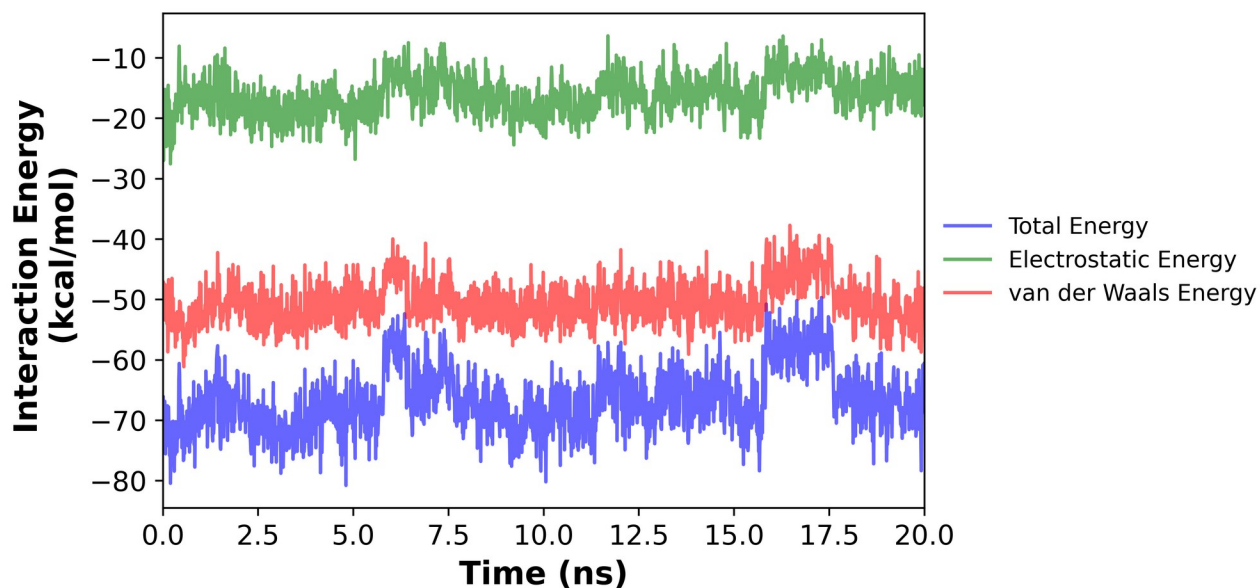
2.4.1. Predictive model novels: figure(4) show the most interaction energy compound in the novels. figure(5) show 30% and 90% novel interactions that occur 30%(1.a, 2.a), 90%(1.b,2.b) for compounds (ZINC000002391833, ZINC000017848357) respectively, and RMSD and 2D RMSD(1.c,2.c) and interaction energy(1.d,2.d) and radius of gyration, detailed information found in supplementary.

2.4.2. RNN model novels: figure(6) show 30% and 90% novel interactions that occur 30%(3.a, 4.a). 90%(3.b,4.b) for compounds (T025V12, T05V56) respectively, and RMSD and 2D RMSD(3.c,4.c) and interaction energy(3.d,4.d) and radius of gyration. detailed information found in supplementary.

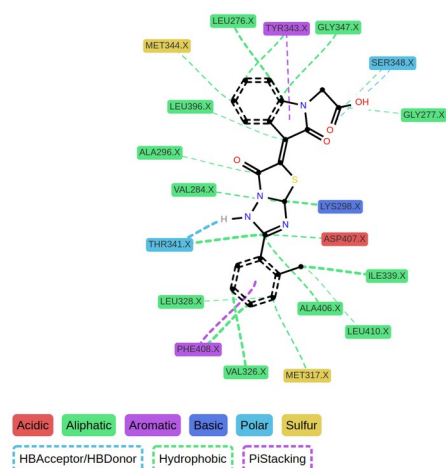
Most compounds have an increasing number of hydrophobic and absence of other interactions like Hydrogen bonds, but it can give insights about the 3D structure or the shape of the ligand and decrease a more steps in the process of fragment-based drug discovery or lead optimization.

All compounds make at least 1 Hydrogen bond and interact with the key residues (ASP407, LYS298, PHE408).

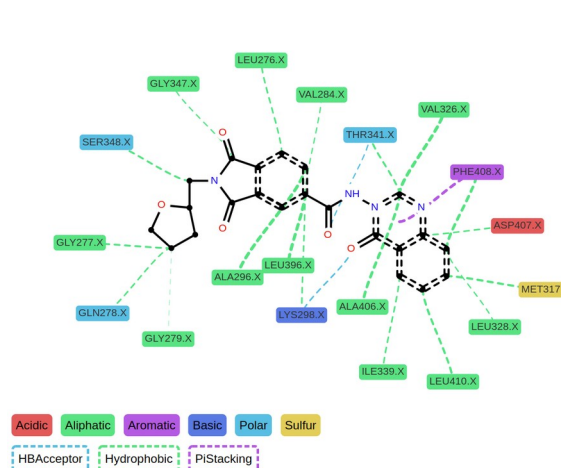
Figure 4: shows interaction energy of novel compound id: ZINC000095478729



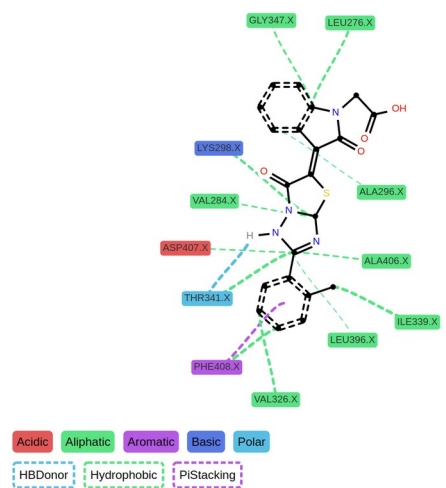
1.a



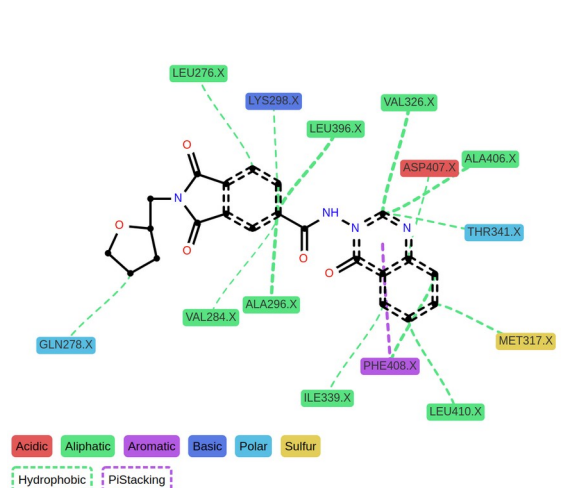
2.a



1.b



2.b



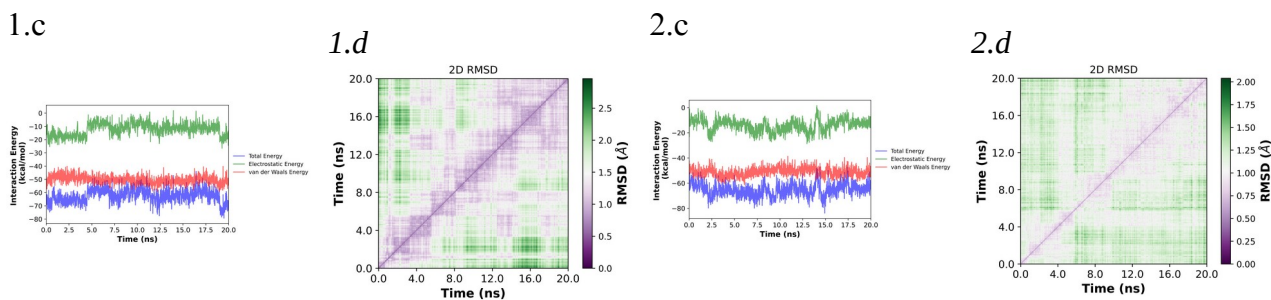
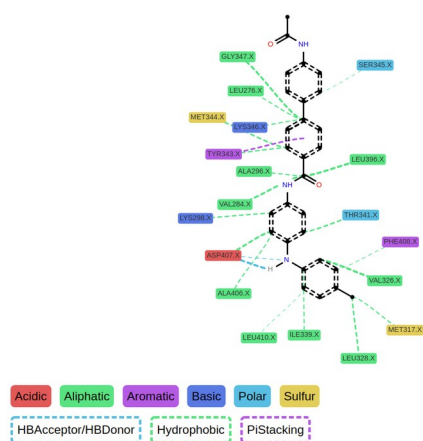
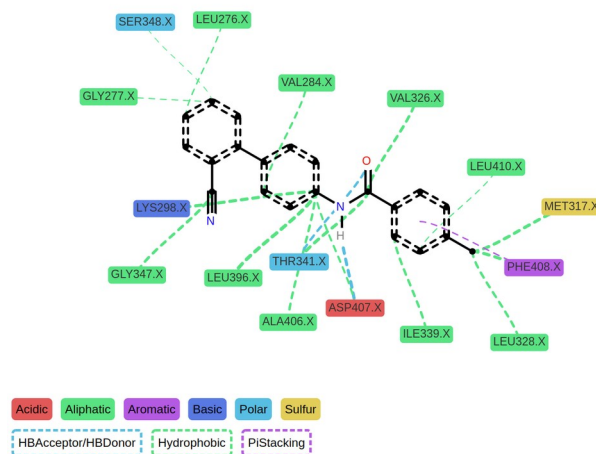


Figure 5: shows novels compounds(1:ZINC00002391833),(2:ZINC000017848357) from ZINC database using predictive model, protein-ligand interaction that occur 30% time (1.a,2.a) and 90% of the time (1.b,2.b) respectively and interaction energy(1.c,2.c) and 2D RMSD (1.d,2.d).

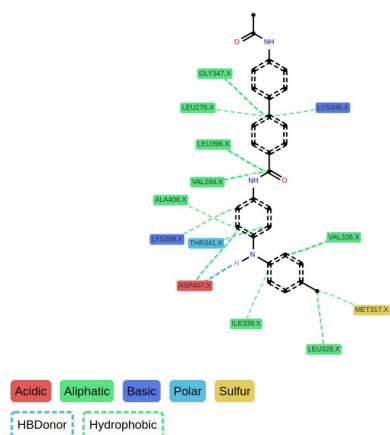
3.a



4.a



3.b



4.b

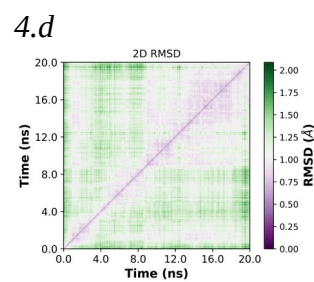
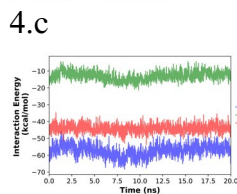
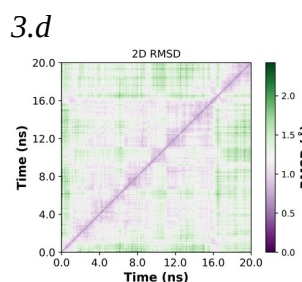
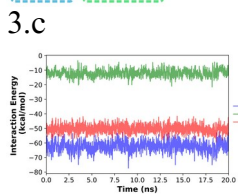
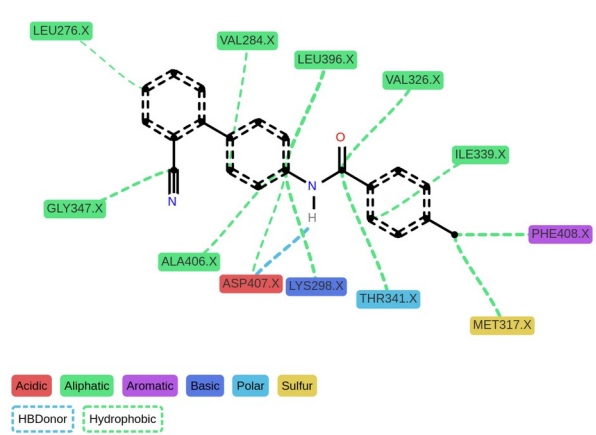


Figure 6: shows novels compounds(3:T025V12), (4:T05V56) from RNN model that used GRU layer and SELFIES, protein-ligand interaction that occur 30% time (3.a,4.a) and 90% of the time (3.b,4.b) respectively and interaction energy(3.c,4.c) and 2D RMSD (3.d,4.d).

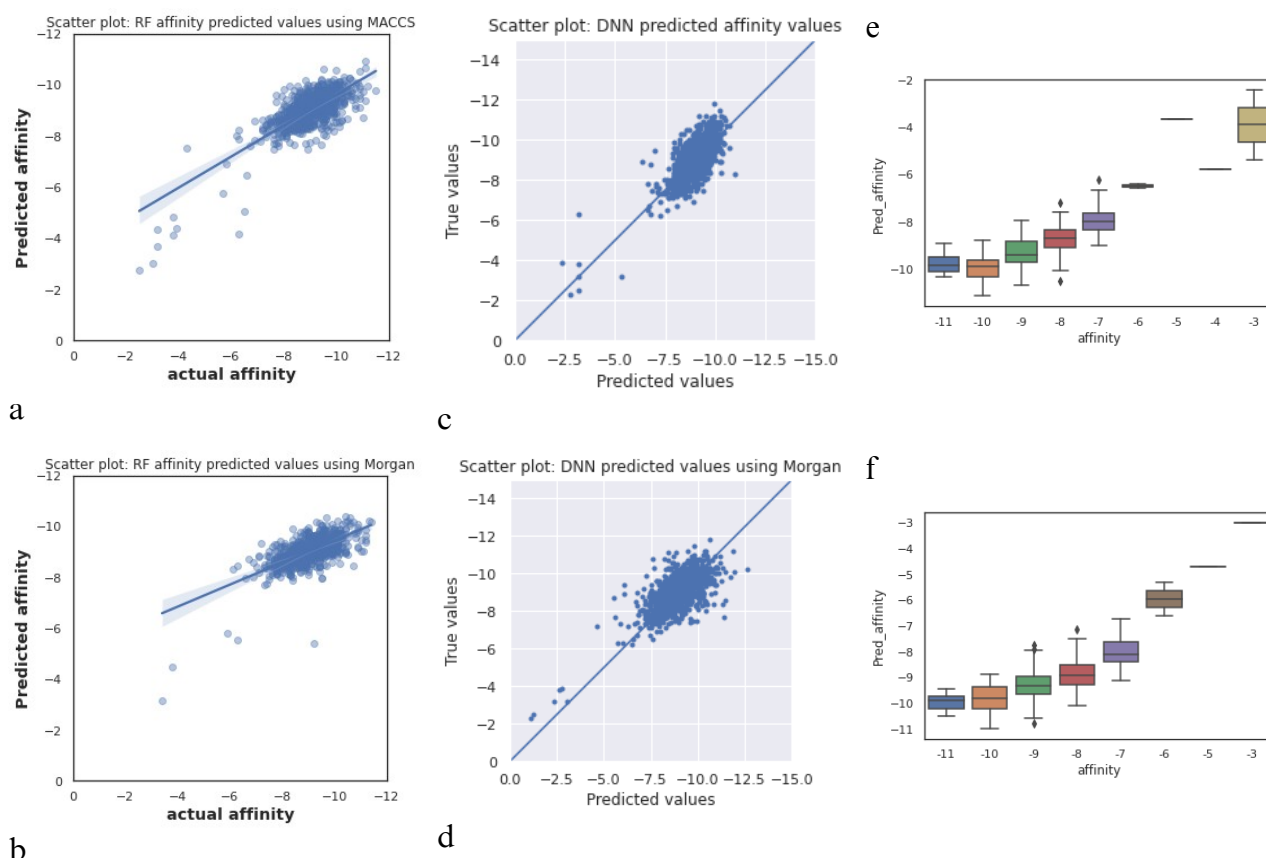


Figure 7: Shows predictive model for docking prediction evaluation (a) RF using MACCS key. (b) RF using Morgan Fingerprint. (c) Deep Neural Networks using MACCS key (d) Deep Neural Networks using Morgan Fingerprint. (e) GCN using SMILES string (f) GCN using Canonical SMILES

4. Docking prediction models: figure(7.a,b) show the machine learning model(Random Forrest) and figure(7.c,d) show Deep learning model both using MACCS Key and Morgan Fingerprint respectively and figure(7.e,f) show GCN model using both SMILES and Canonical smiles. Docking was performed on Data come from 50k ZINC that was used to validate the predictive model. Further docking is recommended as docking data result coming from a certain pocket size and center, however this can make the process fast and give some insight to the user if docking cannot be performed. Mean score of all model can give accurate result.

5. Experimental: Another limitation of the research is that there isn't any experimental evidence of the result, but the further investigation can be done in the future. The Src web app provided with de novo molecular generation by the model give a lot of generated molecules that can be investigated experimentally. Provide a virtual screening ability can guide the experimental process. The study was done for any cancer institute and any research institute that want to provide a cheap and valuable drug that can end the problem of Src kinase targeting and find a final solution for Src kinase.

6. Deployment:

The model deployed on the link provided in GitHub repo (<https://github.com/phalem/Src>) with the Behance link and landing page link. The GitHub link will provide up-to-date information if the link has been changed. Links will be provided in supplementary, but links can be changed in the future.

6.1. Explainable Ai(XAi): The ability to make a prediction explainable is revolutionary and the ability to understand why the model behaves and the reason behind the prediction is a great tool to solve the black-boxing of the model. ExplainableAI(exmol[84] here) show some

limitation due to its principles and the challenge of the ability to make a prediction outside the data, so not all model explains the prediction. It is good to mention that results become time-consuming with a very large descriptors count like modellar (about 15 min. vs 1 min here).

6.2. SaveMol: The website link of SaveMol is found in the GitHub repo (<https://github.com/phalem/SaveMol>). The main app link and Behance link will be on the Github of the app. The ability to store compounds without labeling is a good way to prevent it from stealing, but dealing with a large compound set is a challenge. ReactJS have a problem when dealing with a large list. Some approaches using Windows or other techniques are provided, but the challenge of making a personal database without any limitation on speed will be another challenge. The absence of a model that can be integrated with a program to provide the initial activity prediction of the compound is another limitation, nevertheless it can be provided in the future. KekuleJs sketcher has some limitations in that it can't deal with all SMILES string but it provides a very useful feature due to its open-source nature and is free to use. After all these limitations the app can do its job in the case of RNN molecules that are generated and Scalability and extendability can happen in the future.

## Method:

### 1- Predictive model:

#### 1.1. Data:

1.1.1. Data preparation: the author collect the data with the Uniprot id (P12931) from ChEMBL[89] using ChEMBL web resource client[90] (version 0.10.7), PubChem[91] and DUD-E[92]. Data split into two different types: activity category (active, inactive) and Experimental IC50[93], [94] which is a half maximal inhibitory concentration of a drug or inhibitor. The author use pIC50 a negative log of the IC50 value when converted to molar units as a standard of activity due to IC50 small number that can affect the result and make a balance between active and inactive in case of activity category. DUD-E data was neutralizing using neutralize script[95] that build using RDKit[35].

1.1.2. Molecular Descriptor [17]: the author use MACCS Key fingerprint[18] and Morgan fingerprint[19] as bit vector of 2048 bits integrated into RDKit in both Machine learning model and Deep Neural Network model. In the case of Graph Convolution network the author uses SMILES[36] string and canonical SMILES that are sanitized using RDKit[35] and ConvMolFeaturizer[28] function in deepchem[28] as featurizer.

1.1.3. Data splitting: train\_test\_split function in sklearn[96] was used for splitting, the author split data into 80% train set and 20% test set in case of machine learning and 70% to 30% training set and test set respectively in case of Deep Neural network. The author RandomSplitter function of deepchem[28] to split the data in case of Graph Convolution Network.

#### 1.2. Model training:

1.2.1. Machine learning: the author use MACCS[18] Key and Morgan fingerprints integrated[19] in RDKit[35]. Sklearn[96] library used to perform a model prediction on categorical data using: (1) Support Vector Machine (SVM)[12], [13] with parameter (kernel="rbf", C=1, gamma=0.1, probability=True). (2) Artificial Neural Network (ANN)[14] with parameter (hidden\_layer\_sizes=(30, 3)). (3) Random Forrest (RF) Classifier [11] with parameter (n\_estimators: 100, criterion: entropy). Three-fold cross-validation was performed for all except (SVM in case of morgan fingerprint[19]). In case of Regression Random Forrest (RF) regression (in case of pIC50) with parameter (n\_estimators: 10) was used.

1.2.2. Neural Network: Tensorflow library and Keras [97] was used [24] to build a Neural Network with 2 Dense layer size (64, 32 respectively) using relu activation function and 1 output layer using linear activation function for 50 epochs. Mean square error (mse) as a loss function and adam as optimizer. MACCS Key[18] and morgan[19] was used to make a

regression task on our pIC50 data. In case of MACCs Key the author used batch size 128 and 16 in case of Morgan fingerprints. The hyperparameter chosen coming after plotting loss of every batch size using history technique integrated into Keras and Tensorflow[24].

1.2.3. Graph Neural Network: deepchem Graph Neural network(GraphConvModel function) [28] was used to perform classification and regression tasks with a batch size of 128 with the default parameter in the deepchem model.

1.3. Model Evaluation:

1.3.1. Machine learning: the author calculate sensitivity mean and specificity mean and AUC was used as model Evaluation.

1.3.2. Neural Network: the loss was calculated with mean square error and mean absolute error on a test set. A scatter plot of predicted versus actual value was performed.

1.3.3. Graph Neural Network: Matthews corrccoef mean metric[28] was used as model evaluation in classification task and Accuracy plot performed. Pearson\_r2\_score mean used in prediction and Predicted versus actual (in case of regression) and training versus validation(classification) was plotted using a box plot from the seaborn library[98].

1.4. Model validation and scoring:

1.4.1. Data: The author use a random 50k data from the zinc database[99], remove reactive structure using rd\_filter from '[https://github.com/PatWalters/rd\\_filters](https://github.com/PatWalters/rd_filters)' and remove the compound with certain reactivity and the remaining data was about 39k molecules.

1.4.2. Prediction: the author use predictive bioactivity models to perform prediction and filter it according to every model with HOW\_TO\_VS notebook author used on Github and with paper.

1.4.3. Clustering [15]: after performing prediction and filtering active compounds the author use butina clustering[16] integrated into RDKit[35] to remove redundant data and molecules with the same cluster using cutoff=0.35 and Morgan Fingerprint 3 as 2048 bits.

1.4.4. Scoring:

The presence of crystal structure make it possible to perform structure-based drug discovery using biomolecular simulation and molecular docking[66], the author use the PDB file from the PDB database [100]with protein id number (PDB ID: 7NG7) and resolution 1.5 A and UCW Experimental ligand. The author use SMILES notation[36] as input and open babel pybel[101] to convert SMILES to pdbqt and prepare receptor using mglstools[102]. CBdock[103] was used to make a blind docking[104] and pocket center of (-17.300, -2.055, -5.938) and pocket size of (28.0,28.0,28.0) was given.

After that the author reuse the Volkmer lab[105] helper function that reuse open source tools in a more scripting manner and performs docking using smina[106], [107] which extends autodock vina[108]. After that the author perform protein-ligand interaction using PLIP[70] and its python module[71]. The number of interaction and residues calculated then the author get a mean and max docking for each model and protein residues reached by these model.

1.4.5. Novels: The author choose molecules with high score binding and high hydrogen bond numbers.

1.4.6. Molecular Dynamics[67]: the author use Molecular Dynamics simulation using Making-it-rain[109] default parameter that found in the original GitHub notebooks. The author provides the notebook in the material as in case of any change happen to the original notebook. It uses PyPDB [110], MDTraj[111], PDBFixer[112] and importantly OpenMM library[77] and AMBER[74], [75] force field. The author prepare ligand using general AMBER force field (GAFF[113]) and The Open Force Field Toolkit[114], [115]. The author uses GAFF as a complete force field as it is compatible with the AMBER force field it has parameters for almost all the organic molecules made of C, N, O, H, S, P, F, Cl, Br, and I. LEaP program[116] was used to build a simulation box, and calculate interaction energy according to the complex with a reference ligand that integrated with (PDB id: UCW). Because the author don't use entropy contribution calculation, but true free energy was calculated which can compare similar systems together reference ligand used as a reference to make a comparison. Both the MM-PBSA method and MM-GBSA method were used for comparison. ProLif[117] was used to show the interaction between residue and calculate



interaction bonds and how in percent it occurs throughout molecular dynamics. The author first make molecular dynamics for 5 nanoseconds after that the author extend to 20 nanoseconds for compounds that make more interaction bonds and more energy. Analysis was done using MDanalysis[118], [119] and RMSD was calculated and 2D RMSD and radius of gyration.

## 2- Recurrent Neural Network:

### 2.1.Data:

2.1.1.Data collection: the author collect about random 2 million compounds and 130k in cells and 306k invitro compounds from ZINC[99]database, about 500k compounds from ChEMBL[89] database after that the author filter it using the same rd filter script used before in 50k zinc and remove molecules with reactive structures.

2.1.2.Virtual Screening: the author perform virtual screening using the predictive model that built before and reduce the possibility of active compounds to 500k after that the author get molecules with the most supposed to be active according to our predictive model(threshold 7 for predicted pIC50 or active for classification model) and reduce the data to about 150k compounds. Experimental active molecules that collected before from ChEMBL[89], [90], PubChem[91], and DUD-E[92] is added to 150k compound the model predict and prepare it for model training.

### 2.2.Model training:

2.2.1.Model input: (1)SELFIES[44] was used as the language for model training using the selfies library and used the data that comes from a predictive model which was from ZINC. By training with ZINC, the model training distribution is restricting to molecules that can be synthesized. [nop] was initialized in the list to use for padding as it is a NULL token but for SELFIES[44] Before that, all possible token in the data was extracted and counted. A dictionary was created to make a conversion between string and index and our vocab list. After that, the author will use RNN[37], [44], [62] to predict the whole sequence. A model trained in what is called self-supervised, which masks a part of the data or sequence and gives it to the model to predict the masked part.

(2)SMILES was used as the language for model training and a compound with 2 characters was replaced with a foreign letter(outside SMILES language) like X, Q, etc. The process can be reversed, so 2 character doesn't affect our language and all possible token in the data was extracted and counted. (!) was used as the same rule of [nop] in selfies and was deleted or skipped during evaluation. A dictionary was created to make a conversion between string and index and our vocab list and after that the same process and techniques were used as SELFIES. After evaluation SMILES generated were curated and 2 characters returned in the same manner and SMILES produced were exposed to compound sanitization and evaluated.

2.2.2.Model building: an embedding layer was used and 1 layer of RNN model include 3 different models first one using GRU[62] variant, with return sequences parameter as true and one dense layer using Tensorflow[24] and Keras[97]. Cross entropy as our loss function using SparseCategoricalCrossentropy and from logit parameter equal true using Tensorflow[24] the training was for 4 epochs. Another two models use the same structure but the difference is that the author used the LSTM layer instead of the GRU layer, and the third one uses Bidirectional LSTM instead of GRU. Then for every different model weight was passed to a stateful model to memorize what is fed with the same parameter and the author put its input batch size equal to 1.

2.2.3.Model evaluation: the author gets logit probability and tries to sample from it using 100 molecules from different temperatures (T= 0.15, 0.25, 0.5, 0.75 respectively) to see which one can provide the most inhibitor(Total compound for one structure is 400).

2.2.4.Model scoring: the author used the Structure-based approach that was used before to validate the predictive model, and then calculate the drug score for each molecule using the Volkamer lab helper function[105].

### 2.3.Model Evaluation:

2.3.1.Data filtration: the author filter compounds according to validation, drug-like structure and high-affinity molecules only[66], [69] without protein-ligand interaction consideration[71].  
2.3.2.Molecular Dynamics[67]: after that the author start molecular dynamics simulation in the same steps as in the predictive model with notebooks default parameter as mentioned before.

2.4.Docking prediction model:

2.4.1.Data: the author use the data from the model validation stage of the predictive model and make a regression task for docking values using the same predictive model structures and parameter used before for regression task.

2.4.2.model validation and evaluation: the author make the same metrics as in the Predictive model on regression tasks, with different is that here in data and docking value is the target.

2.4.3.model deployment: the author integrate the docking model with RNN as an assist to make a predictions and scoring with the predictive model.

3. Explainable Ai(XAi)[79]:

3.1.Model Build: the author used exmol[87], [88] to explain prediction to the data. Use MACCS key fingerprint and the Random forest bioactivity classifier that the author developed early. The author uses sample space function from exmol to make an explanation and deploy the model to provide a way to iterate to any model based on SMILES string using streamlit[120] . SMILES input convert to MACCS key and prediction happen on the app.

4. Combine models:

4.1.Explain and predict: the author uses RNN to generate a novel and bioactivity model together with a docking model and calculate Synthetic accessibility (SAS)[34] using sascore.py script and QED[121] using RDKit[35] and give the user chance to discover a new novel molecule, and deployed using streamlit[120].

4.2.Generate and predict and explain: the author combines all things together and generates a molecule using RNN and predicts it using predictive model RF(Random Forrest) and explains it using exmol and deploy all that together to the user to make the process using streamlit[120].

5. Deployment:

5.1.Models: the author use the streamlit library[120] to develop multi-page app and streamlit cloud for model web deployment with a limit number of 100 molecules per prediction (now its 10) and 1 molecule per generation and one molecule per explanation due to lack of funds and limited resources in streamlit[120] with increase number on the future.

5.2.SaveMol:

5.2.1.framework and library: the author used MongoDB[86] as database and Expressjs [85] as backend and React framework with a redux toolkit for Front-end (MERN stack development)[84] to give a full stack platform that can store molecules and use RDKitJs[35] to cure molecules that enter the database. The author handle Exception of Invalid SMILES string that enter or found in database using RDKitJs. Kekulejs[122] provides as a sketcher on the same page to avoid using any outer resource.

5.2.2.Icons and designs: open source 3D icon[123] was used for the main page, but it might change in the future. The author uses blender[124] for building protein and Artificial intelligence icons.

5.2.3.Front-End: the author use React component[84] and make Src prediction landing page that serves as a presentation to explain the steps to build the model and page for sketcher page using kekulejs[122] and one for the main app with substructure search in the front-end using React[84] and RDKit[35].

5.2.4.Backend: the author use nodeJs[125] and its framework Expressjs[85] and provide a hash function for the password using JWT[126] which provide a safe way to save password and mongoose library to deal with MongoDB[86].

5.2.5. Deployment: Heroku app[127] used now to deploy the website on it in case of MERN stack with the aim to change in the future.

## Conclusion:

The ability to generate and predict activity with explanation in a one-page app and store the compound in another app is a very handful tool. The ability to produce a potent inhibitor with a small amount of run and valid molecule generation is a new evolution. Using RNN and SELFIES with Reinforcement learning in the multiobjective optimization can give us a breakthrough in the field without any concern about the validity of the SMILES. Another aspect is the ability and possibility to make the same process for all kinases and thus make it easy to discover selective compounds by providing activity prediction to all. The more automated manner is integrated into a web app and lets serendipity takes place to find selective molecules. Another challenge is to make synthetic accessibility optimization to find an easily synthesizable molecule. The study made the first step to starting all these dreams with the ability to be real. It can be a starting point for anyone who looking for Src kinase inhibitors and to the research institutes that can try those inhibitors experimentally. In the final aspect, I hope this project can decrease the cost of cancer drugs and discover a novel drug that can be made with this free for academic app.

## Abbreviations

RNN: recurrent neural networks

ECFP: Extended connectivity fingerprints

GCN: Graph convolutional network

KNN: K-nearest neighbor

LSTM: Long short-term memory network

MACCS: Molecular ACCess System

RF: Random forest

RMSE: Root mean square error

SMILES: Simplified Molecular Input Line Entry System

SVM: Support vector machine

ANN : Artificial Neural Network

NN: Neural Network

XAI: Explainable Ai

DNN: Deep Neural Networks

VAE: Variational Autoencoder

Ai: Artificial intelligence

MACCS key: Molecular ACCess System keys

ECFPs: extended connectivity fingerprints

GANs: Generated Adversarial Networks

MD :Molecular Dynamics

## Reference:

- [1] P. Cohen, "Protein kinases--the major drug targets of the twenty-first century?," *Nat. Rev. Drug Discov.*, vol. 1, no. 4, pp. 309–315, Apr. 2002, doi: 10.1038/nrd773.
- [2] "The protein kinase complement of the human genome - PubMed." <https://pubmed.ncbi.nlm.nih.gov/12471243/> (accessed Aug. 01, 2022).
- [3] "High-throughput Phenotyping of Lung Cancer Somatic Mutations - PubMed." <https://pubmed.ncbi.nlm.nih.gov/27478040/> (accessed Aug. 01, 2022).
- [4] P. Rous, "A SARCOMA OF THE FOWL TRANSMISSIBLE BY AN AGENT SEPARABLE FROM THE TUMOR CELLS," *J. Exp. Med.*, vol. 13, no. 4, pp. 397–411, Apr. 1911.
- [5] M. A. Seeliger *et al.*, "Equally potent inhibition of c-Src and Abl by compounds that recognize inactive kinase conformations," *Cancer Res.*, vol. 69, no. 6, pp. 2384–2392, Mar. 2009, doi: 10.1158/0008-5472.CAN-08-3953.

- [6] M. Azam, M. A. Seeliger, N. S. Gray, J. Kuriyan, and G. Q. Daley, "Activation of tyrosine kinases by mutation of the gatekeeper threonine," *Nat. Struct. Mol. Biol.*, vol. 15, no. 10, pp. 1109–1118, Oct. 2008, doi: 10.1038/nsmb.1486.
- [7] S. Sosnin, D. Karlov, I. V. Tetko, and M. V. Fedorov, "Comparative Study of Multitask Toxicity Modeling on a Broad Chemical Space," *J. Chem. Inf. Model.*, vol. 59, no. 3, pp. 1062–1072, Mar. 2019, doi: 10.1021/acs.jcim.8b00685.
- [8] Y.-C. Lo, S. E. Rensi, W. Torng, and R. B. Altman, "Machine learning in chemoinformatics and drug discovery," *Drug Discov. Today*, vol. 23, no. 8, pp. 1538–1546, Aug. 2018, doi: 10.1016/j.drudis.2018.05.010.
- [9] A. H. Göller *et al.*, "Bayer's in silico ADMET platform: a journey of machine learning over the past two decades," *Drug Discov. Today*, vol. 25, no. 9, pp. 1702–1709, Sep. 2020, doi: 10.1016/j.drudis.2020.07.001.
- [10] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.
- [11] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [12] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," 2000. doi: 10.1017/CBO9780511801389.013.
- [13] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
- [14] "Frontiers | Editorial: Artificial Neural Networks as Models of Neural Information Processing." <https://www.frontiersin.org/articles/10.3389/fncom.2017.00114/full> (accessed Aug. 01, 2022).
- [15] E. Fix and J. L. Hodges, "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," *Int. Stat. Rev. Rev. Int. Stat.*, vol. 57, no. 3, pp. 238–247, 1989, doi: 10.2307/1403797.
- [16] "Clustering of chemical structures on the basis of two-dimensional similarity measures | Journal of Chemical Information and Modeling." <https://pubs.acs.org/doi/10.1021/ci00010a010> (accessed Aug. 01, 2022).
- [17] "Molecular Descriptors for Chemoinformatics: Volume I: Alphabetical Listing / Volume II: Appendices, References, 2 Volume Set, 2nd, Revised and Enlarged Edition | Wiley." <https://www.wiley.com/en-us/Molecular+Descriptors+for+Chemoinformatics+%3A+Volume+I%3A+Alphabetical+Listing+Volume+II%3A+Appendices%2C+References%2C+2+Volume+Set%2C+2nd%2C+Revised+and+Enlarged+Edition-p-9783527318520> (accessed Aug. 01, 2022).
- [18] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, "Reoptimization of MDL keys for use in drug discovery," *J. Chem. Inf. Comput. Sci.*, vol. 42, no. 6, pp. 1273–1280, Dec. 2002, doi: 10.1021/ci010132r.
- [19] "Extended-Connectivity Fingerprints | Journal of Chemical Information and Modeling." <https://pubs.acs.org/doi/10.1021/ci100050t> (accessed Aug. 01, 2022).
- [20] "RDKit 2012 UGM." <https://www.rdkit.org/UGM/2012/> (accessed Aug. 01, 2022).
- [21] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.
- [22] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking Clever Hans predictors and assessing what machines really learn," *Nat. Commun.*, vol. 10, no. 1, p. 1096, Mar. 2019, doi: 10.1038/s41467-019-08987-4.
- [23] Y. Zhang and A. A. Lee, "Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning," *Chem. Sci.*, vol. 10, no. 35, pp. 8154–8163, Sep. 2019, doi: 10.1039/c9sc00616h.
- [24] T. Developers, "TensorFlow." Zenodo, May 23, 2022. doi: 10.5281/zenodo.6574269.

- [25] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, "Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals," *Chem. Mater.*, vol. 31, no. 9, pp. 3564–3572, May 2019, doi: 10.1021/acs.chemmater.9b01294.
- [26] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *J. Comput. Aided Mol. Des.*, vol. 30, no. 8, pp. 595–608, Aug. 2016, doi: 10.1007/s10822-016-9938-8.
- [27] J. M. Stokes *et al.*, "A Deep Learning Approach to Antibiotic Discovery," *Cell*, vol. 180, no. 4, pp. 688–702.e13, Feb. 2020, doi: 10.1016/j.cell.2020.01.021.
- [28] "Deep Learning for the Life Sciences [Book]." <https://www.oreilly.com/library/view/deep-learning-for/9781492039822/> (accessed Aug. 01, 2022).
- [29] D. Duvenaud *et al.*, "Convolutional Networks on Graphs for Learning Molecular Fingerprints." arXiv, Nov. 03, 2015. doi: 10.48550/arXiv.1509.09292.
- [30] C. M. Dobson, "Chemical space and biology," *Nature*, vol. 432, no. 7019, pp. 824–828, Dec. 2004, doi: 10.1038/nature03192.
- [31] C. Lipinski and A. Hopkins, "Navigating chemical space for biology and medicine," *Nature*, vol. 432, no. 7019, Art. no. 7019, Dec. 2004, doi: 10.1038/nature03193.
- [32] J.-L. Reymond, L. Ruddigkeit, L. Blum, and R. van Deursen, "The enumeration of chemical space," *WIREs Comput. Mol. Sci.*, vol. 2, no. 5, pp. 717–733, 2012, doi: 10.1002/wcms.1104.
- [33] Y. Li, L. Zhang, and Z. Liu, "Multi-objective de novo drug design with conditional graph generative model," *J. Cheminformatics*, vol. 10, no. 1, p. 33, Jul. 2018, doi: 10.1186/s13321-018-0287-6.
- [34] P. Ertl and A. Schuffenhauer, "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions," *J. Cheminformatics*, vol. 1, no. 1, p. 8, Jun. 2009, doi: 10.1186/1758-2946-1-8.
- [35] "RDKit." <https://www.rdkit.org/> (accessed Aug. 01, 2022).
- [36] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, Feb. 1988, doi: 10.1021/ci00057a005.
- [37] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, "Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks," *ACS Cent. Sci.*, vol. 4, no. 1, pp. 120–131, Jan. 2018, doi: 10.1021/acscentsci.7b00512.
- [38] "De Novo Design of Bioactive Small Molecules by Artificial Intelligence - Merk - 2018 - Molecular Informatics - Wiley Online Library." <https://onlinelibrary.wiley.com/doi/10.1002/minf.201700153> (accessed Aug. 01, 2022).
- [39] W. Yuan *et al.*, "Chemical Space Mimicry for Drug Discovery," *J. Chem. Inf. Model.*, vol. 57, no. 4, pp. 875–882, Apr. 2017, doi: 10.1021/acs.jcim.6b00754.
- [40] S. Amabilino, P. Pogány, S. D. Pickett, and D. V. S. Green, "Guidelines for Recurrent Neural Network Transfer Learning-Based Molecular Generation of Focused Libraries," *J. Chem. Inf. Model.*, vol. 60, no. 12, pp. 5699–5713, Dec. 2020, doi: 10.1021/acs.jcim.0c00343.
- [41] "Understanding LSTM Networks -- colah's blog." <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed Aug. 01, 2022).
- [42] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.
- [43] N. O'Boyle and A. Dalke, "DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures," Sep. 2018, doi: 10.26434/chemrxiv.7097960.v1.
- [44] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation," *Mach. Learn. Sci. Technol.*, vol. 1, no. 4, p. 045024, Oct. 2020, doi: 10.1088/2632-2153/aba947.
- [45] S. J. Barigye, J. M. García de la Vega, and Y. Perez-Castillo, "Generative Adversarial Networks (GANs) Based Synthetic Sampling for Predictive Modeling," *Mol. Inform.*, vol. 39, no. 10, p. e2000086, Oct. 2020, doi: 10.1002/minf.202000086.

- [46] Ł. Maziarka, A. Pocha, J. Kaczmarczyk, K. Rataj, T. Danel, and M. Warchoń, “Mol-CycleGAN: a generative model for molecular optimization,” *J. Cheminformatics*, vol. 12, no. 1, p. 2, Jan. 2020, doi: 10.1186/s13321-019-0404-1.
- [47] Y. Khemchandani *et al.*, “DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach,” *J. Cheminformatics*, vol. 12, no. 1, p. 53, Sep. 2020, doi: 10.1186/s13321-020-00454-3.
- [48] T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, and H. Chen, “Application of Generative Autoencoder in De Novo Molecular Design,” *Mol. Inform.*, vol. 37, no. 1–2, p. 1700123, 2018, doi: 10.1002/minf.201700123.
- [49] “Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules | ACS Central Science.” <https://pubs.acs.org/doi/10.1021/acscentsci.7b00572> (accessed Aug. 01, 2022).
- [50] W. Jin, R. Barzilay, and T. Jaakkola, “Junction Tree Variational Autoencoder for Molecular Graph Generation.” arXiv, Mar. 29, 2019. doi: 10.48550/arXiv.1802.04364.
- [51] “[1308.0850] Generating Sequences With Recurrent Neural Networks.” <https://arxiv.org/abs/1308.0850> (accessed Aug. 01, 2022).
- [52] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, “Exploring the Limits of Language Modeling.” arXiv, Feb. 11, 2016. doi: 10.48550/arXiv.1602.02410.
- [53] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [54] B. Sanchez-Lengeling, C. Outeiral, G. L. Guimaraes, and A. Aspuru-Guzik, “Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC),” Aug. 2017, doi: 10.26434/chemrxiv.5309668.v3.
- [55] R. Winter, F. Montanari, A. Steffen, H. Briem, F. Noé, and D.-A. Clevert, “Efficient Multi-Objective Molecular Optimization in a Continuous Latent Space,” Apr. 2019, doi: 10.26434/chemrxiv.7971101.v1.
- [56] M. Hartenfeller *et al.*, “DOGS: Reaction-Driven de novo Design of Bioactive Compounds,” *PLOS Comput. Biol.*, vol. 8, no. 2, p. e1002380, Feb. 2012, doi: 10.1371/journal.pcbi.1002380.
- [57] T. Unterthiner, A. Mayr, G. Klambauer, and S. Hochreiter, “Toxicity Prediction using Deep Learning.” arXiv, Mar. 04, 2015. doi: 10.48550/arXiv.1503.01445.
- [58] S. Kearnes, B. Goldman, and V. Pande, “Modeling Industrial ADMET Data with Multitask Networks.” arXiv, Jan. 12, 2017. doi: 10.48550/arXiv.1606.08793.
- [59] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen, “Molecular de-novo design through deep reinforcement learning,” *J. Cheminformatics*, vol. 9, no. 1, p. 48, Sep. 2017, doi: 10.1186/s13321-017-0235-x.
- [60] T. Blaschke, O. Engkvist, J. Bajorath, and H. Chen, “Memory-assisted reinforcement learning for diverse molecular de novo design,” *J. Cheminformatics*, vol. 12, no. 1, p. 68, Nov. 2020, doi: 10.1186/s13321-020-00473-0.
- [61] M. Popova, O. Isayev, and A. Tropsha, “Deep reinforcement learning for de novo drug design,” *Sci. Adv.*, vol. 4, no. 7, p. eaap7885, Jul. 2018, doi: 10.1126/sciadv.aap7885.
- [62] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” arXiv, Sep. 02, 2014. doi: 10.48550/arXiv.1406.1078.
- [63] M. Johnson *et al.*, “Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation.” arXiv, Aug. 21, 2017. Accessed: Aug. 01, 2022. [Online]. Available: <http://arxiv.org/abs/1611.04558>
- [64] “Molecular Design: Concepts and Applications | Wiley.” <https://www.wiley.com/en-us/Molecular+Design%3A+Concepts+and+Applications-p-9783527314324> (accessed Aug. 01, 2022).

- [65] D. Stumpfe and J. Bajorath, "Similarity searching," *WIREs Comput. Mol. Sci.*, vol. 1, no. 2, pp. 260–282, 2011, doi: 10.1002/wcms.23.
- [66] D. B. Kitchen, H. Decornez, J. R. Furr, and J. Bajorath, "Docking and scoring in virtual screening for drug discovery: methods and applications," *Nat. Rev. Drug Discov.*, vol. 3, no. 11, pp. 935–949, Nov. 2004, doi: 10.1038/nrd1549.
- [67] J. D. Durrant and J. Mccammon, "Molecular dynamics simulations and drug discovery," *BMC Biol.*, 2011, doi: 10.1186/1741-7007-9-71.
- [68] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui, "Molecular docking: a powerful approach for structure-based drug discovery," *Curr. Comput. Aided Drug Des.*, vol. 7, no. 2, pp. 146–157, Jun. 2011, doi: 10.2174/157340911795677602.
- [69] X. Du *et al.*, "Insights into Protein–Ligand Interactions: Mechanisms, Models, and Methods," *Int. J. Mol. Sci.*, vol. 17, no. 2, Art. no. 2, Feb. 2016, doi: 10.3390/ijms17020144.
- [70] S. Salentin, S. Schreiber, V. J. Haupt, M. F. Adasme, and M. Schroeder, "PLIP: fully automated protein–ligand interaction profiler," *Nucleic Acids Res.*, vol. 43, no. W1, pp. W443–W447, Jul. 2015, doi: 10.1093/nar/gkv315.
- [71] M. F. Adasme *et al.*, "PLIP 2021: expanding the scope of the protein–ligand interaction profiler to DNA and RNA," *Nucleic Acids Res.*, vol. 49, no. W1, pp. W530–W534, Jul. 2021, doi: 10.1093/nar/gkab294.
- [72] M. Christen *et al.*, "The GROMOS software for biomolecular simulation: GROMOS05," *J. Comput. Chem.*, vol. 26, no. 16, pp. 1719–1751, Dec. 2005, doi: 10.1002/jcc.20303.
- [73] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, "CHARMM: A program for macromolecular energy, minimization, and dynamics calculations," *J. Comput. Chem.*, vol. 4, no. 2, pp. 187–217, 1983, doi: 10.1002/jcc.540040211.
- [74] W. D. Cornell *et al.*, "A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules," *J. Am. Chem. Soc.*, vol. 117, no. 19, pp. 5179–5197, May 1995, doi: 10.1021/ja00124a002.
- [75] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case, "Development and testing of a general amber force field," *J. Comput. Chem.*, vol. 25, no. 9, pp. 1157–1174, 2004, doi: 10.1002/jcc.20035.
- [76] "GitHub - openmm/openmmforcefields: CHARMM and AMBER forcefields for OpenMM (with small molecule support)." <https://github.com/openmm/openmmforcefields> (accessed Aug. 01, 2022).
- [77] "openmm/openmm." OpenMM, Jul. 30, 2022. Accessed: Aug. 01, 2022. [Online]. Available: <https://github.com/openmm/openmm>
- [78] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR." arXiv, Mar. 21, 2018. doi: 10.48550/arXiv.1711.00399.
- [79] J. Jiménez-Luna, F. Grisoni, and G. Schneider, "Drug discovery with explainable artificial intelligence," *Nat. Mach. Intell.*, vol. 2, no. 10, Art. no. 10, Oct. 2020, doi: 10.1038/s42256-020-00236-4.
- [80] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 427–436. doi: 10.1109/CVPR.2015.7298640.
- [81] S. Riniker and G. A. Landrum, "Similarity maps - a visualization strategy for molecular fingerprints and machine-learning methods," *J. Cheminformatics*, vol. 5, no. 1, p. 43, Sep. 2013, doi: 10.1186/1758-2946-5-43.
- [82] D. Numeroso and D. Bacciu, "Explaining Deep Graph Networks with Molecular Counterfactuals." arXiv, Nov. 09, 2020. doi: 10.48550/arXiv.2011.05134.
- [83] "JavaScript | MDN." <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (accessed Aug. 01, 2022).

- [84] “React – A JavaScript library for building user interfaces.” <https://reactjs.org/> (accessed Aug. 01, 2022).
- [85] “Express - Node.js web application framework.” <https://expressjs.com/> (accessed Aug. 01, 2022).
- [86] “MongoDB: The Developer Data Platform | MongoDB | MongoDB.” <https://www.mongodb.com/> (accessed Aug. 01, 2022).
- [87] G. P. Wellawatte, A. Seshadri, and A. D. White, “Model agnostic generation of counterfactual explanations for molecules,” *Chem. Sci.*, vol. 13, no. 13, pp. 3697–3705, Mar. 2022, doi: 10.1039/D1SC05259D.
- [88] H. A. Gandhi and A. D. White, “Explaining structure-activity relationships using locally faithful surrogate models,” May 2022, doi: 10.26434/chemrxiv-2022-v5p6m-v2.
- [89] A. Gaulton *et al.*, “The ChEMBL database in 2017,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D945–D954, Jan. 2017, doi: 10.1093/nar/gkw1074.
- [90] “ChEMBL webresource client.” The ChEMBL Group, Jul. 18, 2022. Accessed: Aug. 01, 2022. [Online]. Available: [https://github.com/chembl/chembl\\_webresource\\_client](https://github.com/chembl/chembl_webresource_client)
- [91] S. Kim *et al.*, “PubChem Substance and Compound databases,” *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1202–1213, Jan. 2016, doi: 10.1093/nar/gkv951.
- [92] M. M. Mysinger, M. Carchia, John. J. Irwin, and B. K. Shoichet, “Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking,” *J. Med. Chem.*, vol. 55, no. 14, pp. 6582–6594, Jul. 2012, doi: 10.1021/jm300687e.
- [93] B. Beck *et al.*, “Assay Operations for SAR Support,” in *Assay Guidance Manual*, S. Markossian, A. Grossman, K. Brimacombe, M. Arkin, D. Auld, C. Austin, J. Baell, T. D. Y. Chung, N. P. Coussens, J. L. Dahlin, V. Devanarayan, T. L. Foley, M. Glicksman, J. V. Haas, M. D. Hall, S. Hoare, J. Inglese, P. W. Iversen, S. C. Kales, M. Lal-Nag, Z. Li, J. McGee, O. McManus, T. Riss, P. Saradjian, G. S. Sittampalam, M. Tarselli, O. J. Trask, Y. Wang, J. R. Weidner, M. J. Wildey, K. Wilson, M. Xia, and X. Xu, Eds. Bethesda (MD): Eli Lilly & Company and the National Center for Advancing Translational Sciences, 2004. Accessed: Aug. 01, 2022. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK91994/>
- [94] S. Offermanns and W. Rosenthal, Eds., “IC50 Values,” in *Encyclopedia of Molecular Pharmacology*, Berlin, Heidelberg: Springer, 2008, pp. 611–611. doi: 10.1007/978-3-540-38918-7\_5943.
- [95] “DeepLearningLifeSciences.” deepchem, Jul. 29, 2022. Accessed: Aug. 01, 2022. [Online]. Available: <https://github.com/deepchem/DeepLearningLifeSciences/blob/9020c18d97de5f5bdab85234a5f3aac191791f1e/Chapter11/neutralize.py>
- [96] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Mach. Learn. PYTHON*, p. 6.
- [97] “Keras: Deep Learning for humans.” Keras, Aug. 01, 2022. Accessed: Aug. 01, 2022. [Online]. Available: <https://github.com/keras-team/keras>
- [98] M. L. Waskom, “seaborn: statistical data visualization,” *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, Apr. 2021, doi: 10.21105/joss.03021.
- [99] T. Sterling and J. J. Irwin, “ZINC 15 – Ligand Discovery for Everyone,” *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, Nov. 2015, doi: 10.1021/acs.jcim.5b00559.
- [100] R. P. D. Bank, “RCSB PDB: Homepage.” <https://www.rcsb.org/> (accessed Aug. 01, 2022).
- [101] “Open Babel.” [http://openbabel.org/wiki/Main\\_Page](http://openbabel.org/wiki/Main_Page) (accessed Aug. 01, 2022).
- [102] mgl-admin, “Downloads,” *mgltools*. <https://ccsb.scripps.edu/mgltools/downloads/> (accessed Aug. 01, 2022).
- [103] Y. Liu, M. Grimm, W. Dai, M. Hou, Z.-X. Xiao, and Y. Cao, “CB-Dock: a web server for cavity detection-guided protein–ligand blind docking,” *Acta Pharmacol. Sin.*, vol. 41, no. 1, pp. 138–144, Jan. 2020, doi: 10.1038/s41401-019-0228-6.



- [104] N. M. Hassan, A. A. Alhossary, Y. Mu, and C.-K. Kwok, "Protein-Ligand Blind Docking Using QuickVina-W With Inter-Process Spatio-Temporal Integration," *Sci. Rep.*, vol. 7, no. 1, Art. no. 1, Nov. 2017, doi: 10.1038/s41598-017-15571-7.
- [105] D. Sydow, A. Morger, M. Driller, and A. Volkamer, "TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data," *J. Cheminformatics*, vol. 11, no. 1, p. 29, Apr. 2019, doi: 10.1186/s13321-019-0351-x.
- [106] "smina download | SourceForge.net." <https://sourceforge.net/projects/smina/> (accessed Aug. 01, 2022).
- [107] D. R. Koes, M. P. Baumgartner, and C. J. Camacho, "Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise," *J. Chem. Inf. Model.*, vol. 53, no. 8, pp. 1893–1904, Aug. 2013, doi: 10.1021/ci300604z.
- [108] "AutoDock Vina." <https://vina.scripps.edu/> (accessed Aug. 01, 2022).
- [109] P. R. Arantes, M. D. Polêto, C. Pedebos, and R. Ligabue-Braun, "Making it Rain: Cloud-Based Molecular Simulations for Everyone," *J. Chem. Inf. Model.*, vol. 61, no. 10, pp. 4852–4856, Oct. 2021, doi: 10.1021/acs.jcim.1c00998.
- [110] W. Gilpin, "PyPDB." Aug. 01, 2022. Accessed: Aug. 01, 2022. [Online]. Available: <https://github.com/williamgilpin/pypdb>
- [111] "mdtraj/mdtraj." MDTraj, Jul. 24, 2022. Accessed: Aug. 01, 2022. [Online]. Available: <https://github.com/mdtraj/mdtraj>
- [112] "PDBFixer." OpenMM, Aug. 01, 2022. Accessed: Aug. 01, 2022. [Online]. Available: <https://github.com/openmm/pdbfixer>
- [113] "GAFF." <http://ambermd.org/antechamber/gaff.html> (accessed Aug. 01, 2022).
- [114] D. L. Mobley *et al.*, "Open Force Field Consortium: Escaping atom types using direct chemical perception with SMIRNOFF v0.1." bioRxiv, p. 286542, Jul. 13, 2018. doi: 10.1101/286542.
- [115] J. Wagner *et al.*, "openforcefield/openff-toolkit: 0.10.3 Bugfix release." Zenodo, Feb. 28, 2022. doi: 10.5281/zenodo.6310995.
- [116] "Fundamentals of LEaP." <https://ambermd.org/tutorials/pengfei/index.php> (accessed Aug. 02, 2022).
- [117] "ProLIF: a library to encode molecular interactions as fingerprints | Journal of Cheminformatics | Full Text." <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-021-00548-6> (accessed Aug. 02, 2022).
- [118] R. J. Gowers *et al.*, "MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations," *Proc. 15th Python Sci. Conf.*, pp. 98–105, 2016, doi: 10.25080/Majora-629e541a-00e.
- [119] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations," *J. Comput. Chem.*, vol. 32, no. 10, pp. 2319–2327, Jul. 2011, doi: 10.1002/jcc.21787.
- [120] "GitHub - streamlit/streamlit: Streamlit — The fastest way to build data apps in Python." <https://github.com/streamlit/streamlit> (accessed Aug. 02, 2022).
- [121] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins, "Quantifying the chemical beauty of drugs," *Nat. Chem.*, vol. 4, no. 2, pp. 90–98, Jan. 2012, doi: 10.1038/nchem.1243.
- [122] C. Jiang, X. Jin, Y. Dong, and M. Chen, "Kekule.js: An Open Source JavaScript Cheminformatics Toolkit," *J. Chem. Inf. Model.*, vol. 56, no. 6, pp. 1132–1138, Jun. 2016, doi: 10.1021/acs.jcim.6b00167.
- [123] "3dicons - Open source 3D icon library." <https://3dicons.co/> (accessed Aug. 02, 2022).
- [124] "Community — blender.org." <https://www.blender.org/community/> (accessed Aug. 01, 2022).
- [125] Node.js, "Node.js," *Node.js*. <https://nodejs.org/en/> (accessed Aug. 02, 2022).
- [126] auth0.com, "JWT.IO." <http://jwt.io/> (accessed Aug. 02, 2022).
- [127] "Cloud Application Platform | Heroku." <https://www.heroku.com/> (accessed Aug. 02, 2022).