Machine learning methods for drug-target binding affinity prediction

Εθνικό Μετσόβιο Πολυτεχνείο

Εργαστήριο Βιοϊατρικών Συστημάτων

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΏΝ ΚΑΤΑΣΚΕΥΩΝ

Διπλωματική Εργασία

Φοιτητής: Θωμάς Γκέκας

Επιβλέπων: Αλεξόπουλος Λεωνίδας

Αναπληρωτής Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2020

Machine learning methods for drug-target binding affinity prediction

Εθνικό Μετσόβιο Πολυτεχνείο

Εργαστήριο Βιοϊατρικών Συστημάτων

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΏΝ ΚΑΤΑΣΚΕΥΩΝ

Διπλωματική Εργασία

Φοιτητής: Θωμάς Γκέκας

Επιβλέπων: Αλεξόπουλος Λεωνίδας

Αναπληρωτής Καθηγητής ΕΜΠ

Εγκρίθηκε την 16η Οκτωβρίου 2020 από την τριμελή επιτροπή:

Αλεξόπουλος Λεωνίδας	Προβατίδης Χριστόφορος	Κυριακόπουλος Κωνσταντίνος
Αναπληρωτής Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2020

Abstract

Measuring the binding affinity from ligand-protein interactions is one of the most important stages at drug development and drug design. Computational drug discovery has rapidly evolved nowadays, decreasing significantly trials' time and number of new compounds, by deducting those that have computationally been rejected. For that reason, in this diploma thesis, some computational models will be suggested based on machine learning and especially on deep learning, which will be able to learn major features from the drugs' chemical structures, in order to predict with high accuracy the binding affinity with specific proteins. Deep learning models will consist of graph convolutional networks, that are responsible for encoding the chemical structures and extract features from them. Moreover, methods like multitask or one-shot learning will be implemented due to high efficiency and performance to similar applications. Eventually, the aim is to improve the generalization performance with new methods in comparison with older models.

Περίληψη

Ο υπολογισμός της δύναμης ζεύξης από τις αντιδράσεις μεταξύ ενός φαρμάκου και μίας πρωτεΐνης αποτελεί ένα από τα πιο σημαντικά στάδια στην ανάπτυξη και σχεδιασμό φαρμάκων. Ο υπολογιστικός σχεδιασμός φαρμάκων έχει αναπτυχθεί ραγδαία τα τελευταία χρόνια, μειώνοντας σημαντικά τον χρόνο και τον όγκο δοκιμών σε νέες δομές φαρμάκων, αφαιρώντας αυτές που έχουν αποδειχθεί υπολογιστικά ως μη αποδεκτές. Για αυτό το λόγο, σε αυτή τη διπλωματική εργασία, θα προταθούν υπολογιστικά μοντέλα βασισμένα στη μηχανική μάθηση και συγκεκριμένα στη βαθεία μάθηση, τα οποία θα είναι ικανά να μαθαίνουν βασικά χαρακτηριστικά από τις χημικές δομές των φαρμάκων ούτως ώστε να προβλέπουν με υψηλή ακρίβεια τη δύναμη ζεύξης τους με ορισμένες πρωτεΐνες. Τα μοντέλα βαθείας μάθησης θα αποτελούνται από συνελικτικά δίκτυα γράφων, τα οποία είναι υπεύθυνα για την κωδικοποίηση των χημικών τύπων των μορίων φαρμάκων και για την σωστή εξαγωγή χαρακτηριστικών αυτών. Ακόμη, μέθοδοι όπως η εκμάθηση πολλών καθηκόντων και η one-shot τεχνική θα εφαρμοσθούν εξαιτίας της υψηλής τους αποτελεσματικότητας και απόδοσης σε παρόμοιες εφαρμογές. Τελικώς, ο στόχος είναι η βελτίωση της απόδοσης των μοντέλων ανεξαρτήτως δυσκολίας και διαχωρισμό των δεδομένων με νέες μεθόδους σε σχέση με παλιότερες μελέτες.

Acknowledgements

At first, i would like to thank my supervisor Leonidas Alexopoulos for accepting me as his laboratory's member and giving me the opportunity to improve my knowledge about bioinformatics and data science. Also, i would like to thank all of my lab mates and friends, Nikos, Giorgos, Kostas, Danai, Nagia and Mari and especially my supervisor's PhD student, Christos Fotis, who was a project partner and a mentor to me. He helped me to improve my skills and complete my thesis despite his full-time work timeline and other projects he worked on. Last but not least, i want to thank my family, my parents for supporting me all this years to help me study at university and my sisters.

Table of Contents

List of Figures

List of Tables

1 | Introduction

1.1 Motivation and drug discovery

Drug Discovery is the process of developing a new drug/compound to use as a treatment or cure. It involves many scientific fields like biology, chemistry, pharmacology, and nowadays computer science.

Traditional drug discovery consists of experimental trials in a laboratory environment with the most known and usable technique being High-throughput screening (HTS). HTS allows conducting millions of chemical, genetic, or pharmacological experiments and to identify active compounds, antibodies, or genes that modulate a particular biomolecular pathway. However, despite the big amount of compounds that are tested, it takes a lot of time and preparation to build biological assays and then conduct the experiment, while most compounds are inactive. That means that only a few compounds will proceed to the next stage of drug design.

On the other hand, the last decade, Computer-aided Drug Discovery (CADD) is used as a complementary process of drug design, not as a replacement though, of HTS (Sliwoski, Kothiwale, Meiler, & Edward W. Lowe, 2014). In fact, CADD tools are improving HTS efficiency, because initial screening is conducted and so only compounds that are computationally identified as actives will be prioritized for further experiments decreasing, by far the preparation and execution time of screening. Computational drug discovery, as shown in Figure 1 is classified into two categories: structure-based and ligand-based.

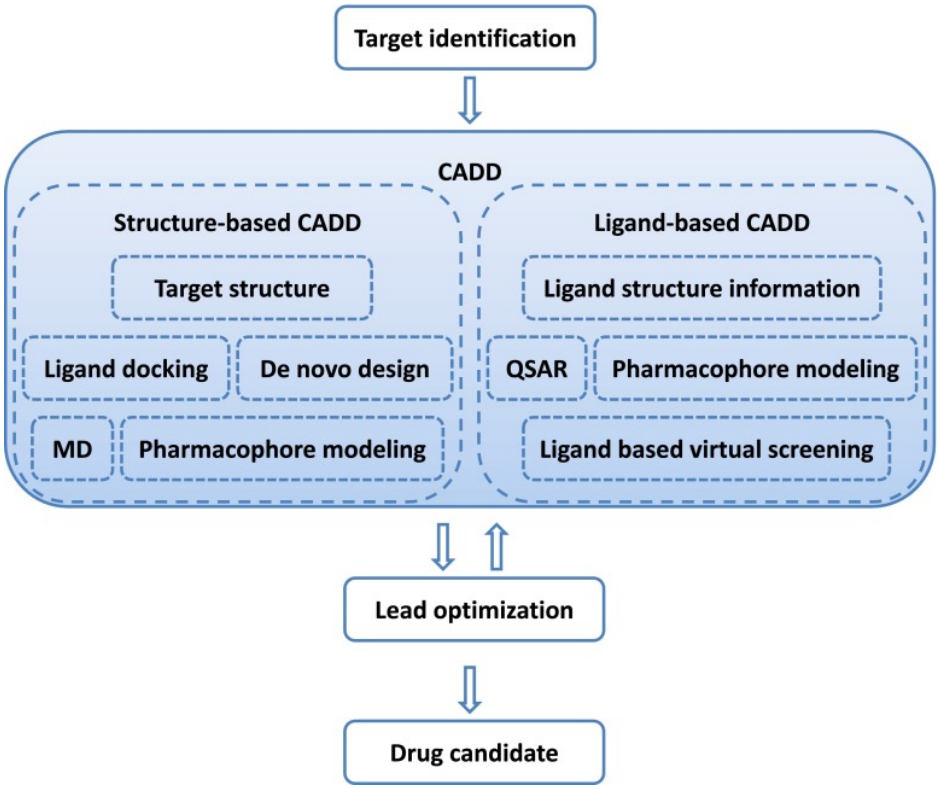

Figure 1: Computer-aided drug discovery methods.

Structure-based CADD relies on the knowledge of the target protein structure to calculate interaction energies for all compounds tested, whereas ligand-based exploits the knowledge of known active and inactive molecules through chemical similarity searches or construction of predictive, quantitative structure-activity relation models. For this thesis purposes, a ligand-based approach was chosen because of the available data that will be analyzed in Chapter 2.

1.2 Binding affinity

Binding affinity is the strength of the binding interactions between a single biomolecule, for example, a protein, to its ligand partner e.g a drug. It is typi-

cally measured and reported by the equilibrium dissociation constant K_D which is used to evaluate and rank order strengths of bio-molecular interactions. Binding strength depends on the constant's value because a smaller value means greater binding affinity of the drug for its target and on the other hand larger K_D describes weaker interactions between the biomolecule and the drug. Binding affinity is affected by non-covalent intermolecular interactions such as hydrogen bonding, electrostatic interactions, hydrophobic and Van der Waals forces between the drug-target pair.

Figure 2: Binding process.

In drug discovery, binding affinity is usually utilized for the effectiveness of a new compound which is at its first stages of development. For example, a new drug is aiming at a determinate disease or cure to inhibit a molecular target central that corresponds to the disease's mechanism of interest. These target centrals are proteins in human cells. Therefore, to accomplish the inhibition of the target demands successful binding between the drug and the protein. However, K_D constant is not the only metric to measure the binding affinity because of experiment complexity. The half-maximal inhibitory concentration, IC_{50}, is also another solution of binding affinity measurement. That constant describes the effectiveness of a substance in inhibiting a specific biological or biochemical function. In binding affinity, just as K_D, smaller IC_{50} value is associated with stronger drug-target interactions

and vice versa which allows using this constant due to easier and more accurate measurements from in-vitro experiments.

1.3 Machine learning

Artificial Intelligence (AI) is commonly used nowadays for many applications such as face and voice recognition, financial management, control systems, due to the rapid increase of computational power. Machine Learning algorithms are implemented more efficiently giving results faster than the human brain while helping to solve problems that were not able to calculate by hand. Generally, a machine learning model is like a black box (Figure 3) with inputs and outputs, where inputs are some data and outputs are something that its developer wants to predict.

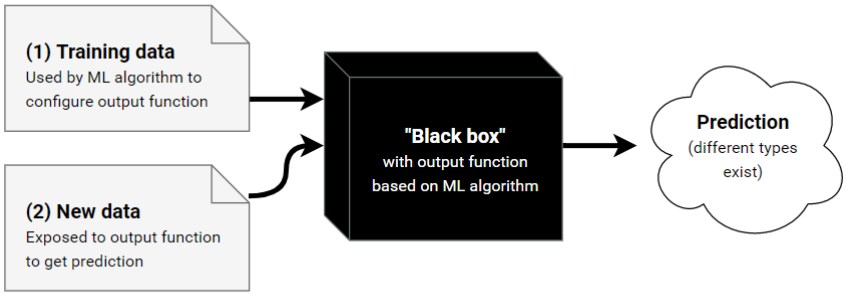

Figure 3: Machine learning model

However, the true power of that model is the potential it has to be trained from that data to predict a specific value for other, different data with high accuracy. For example, an ML model could have as inputs animal images and as outputs the kind of the animal which is a very common classification problem. So, that black box actually during the training is building a function that can be used to predict

a particular target.

Particularly machine learning consists of many different kinds of algorithms with the most known of them to be Deep Learning, which is used for this thesis purposes too. Deep Learning is based on the human biological neurons and their ability to transmit information through an electrochemical process. Consequently, an artificial neural network (ANN) was invented inspired by physical neurons in order to learn an objective from some data. A simple ANN is constructed from a batch of layers (n<3 with n the number of layers), each of them includes a number of neurons, which transfer the information of the input data from layer to layer mapping them on a non-linear function until the final that is the output of the model and specific value to be predicted. Moreover, passing through the data from an ANN is called training and corresponds from training parameters, or else weights of each neuron's function, in order to minimize the error of the predicted values. Finally, to train an ANN is necessary to assign a cost or an objective function between the output and the true values.

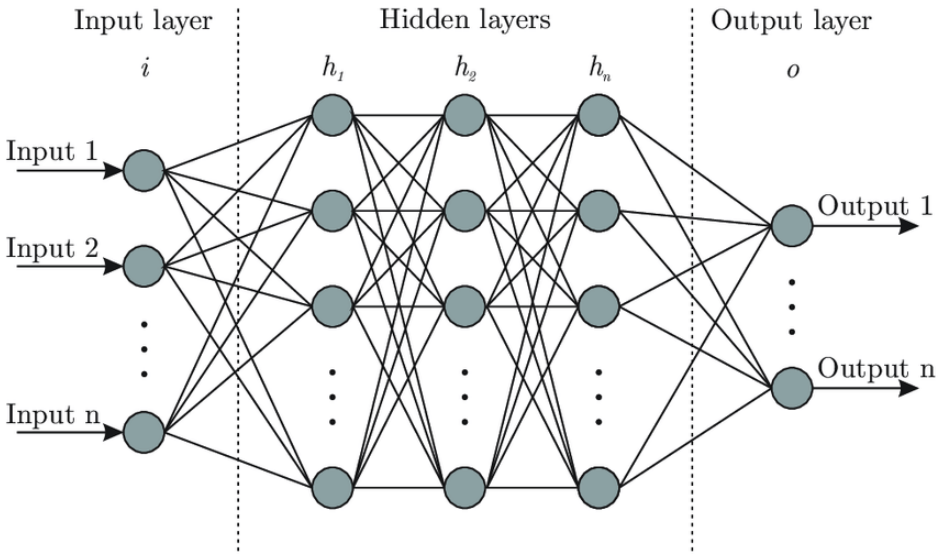

Figure 4: Deep neural network

A more complicated version of ANN is called Deep Neural Network (DNN). DNNs are the same as ANNs with the difference that there are more layers between the input and the output, which are called hidden layers, that gives the capability of learning features from raw data due to the increased number of neurons and so the number of trainable parameters.

1.4 Deep learning in binding affinity

In the binding affinity topic, there is already progress towards finding the solution of predicting dissociation constant K_D with deep learning methods. One of the first attempts is DeepDTA (Öztürk, Özgür, & Ozkirimli, 2018). The work of the DeepDTA model concerns the modeling of protein sequences and compound 1D representations with convolutional neural networks (CNNs). The paper that was published in June 2018 managed to overpass the previous baselines that were the KronRLS regression algorithm (Pahikkala et al., 2014) and the SimBoost method (He, Heidemeyer, Ban, Cherkasov, & Ester, 2017).

Next is PADME (Feng, Dueva, Cherkasov, & Ester, 2018), a deep learning-based DTI prediction model which uses the combined smallmolecule compound (candidate drug) and target protein feature vectors. It considers two variants of PADME with either Molecular Graph Convolution (MGC) or ECFP as the compound featurization method. For the protein, they uses Protein Sequence Composition (PSC) descriptor. The compound vector is concatenated with a target protein vector to form the Combined Input Vector (CIV) for the neural network. PADME predicts a real-valued interaction strength.

In addition, another computational model on binding affinity is GraphDTA (Nguyen, Le, & Venkatesh, 2019). This deep learning model includes two graph convolutional networks, one for the compounds and one for targets. In the beginning, the

molecular features of compounds, are converted into molecular graphs and then passed through the GCN and moreover proteins sequences are transformed to embedding vectors, imported to the other GCN until both of them concatenated to one input for the deep neural network in order to predict binding affinity from the output layer.

All of these models were aiming to predict the K_D value for the drug-target pairs based on regression analysis. However, these methods did not demonstrate high performance and, conclusively, further research is needed, towards different techniques to increase accuracy on predicted values.

1.5 Current study

In the current diploma thesis, several machine learning methods are discussed, with the purpose of predicting binding affinity for drug-target interactions such as the previous in section 1.4. However, the aim in the current study is to classify the compound-protein pairs according to the binding capability, meaning the ability of a drug-molecule to bind or not with a specific protein, changing the problem from a regression problem to classification one.

Moreover, despite the amount of data given for K_D, IC_{50} values were selected from the data to be used as the metric determining the classification of a drug as a binder or not. Finally, another difference from the previous models is that this thesis' methods are target-oriented, meaning that every model is trained to predict binding among one protein. This decreases the total amount of data due to fewer target-specific pairs of drug-protein, but, at the same time, it seems to give better results because the model is trained only on compounds, making it easier to understand the chemical features of their structures.

Therefore, there is the base model which is a combination of graph convolution

and deep neural network, constructed to predict the labels of the drugs and additionally variations of that model according to other techniques that will be reported in Chapter 3. Also, a gradient boosting model (XGBoost) will be used as a stand-alone and an evaluation model and some baseline methods such as kNN and random forest for comparison purposes.

2 | Data

2.1 Raw data

At first, the data that used for this project was gathered from the IDG-DREAM Drug-Kinase Binding Prediction Challenge (Cichonska et al., 2020) which was also about binding affinity problem. It contains almost 5 million data points, but not all of them useful, with information from experimental processes measuring different bio-activities such as dissociation constant, IC_{50}, and other binding related constants, for pairs of several drug compounds and proteins.

2.2 Data preprocessing

2.2.1 Target selection

Despite the big number of data points, it is important to filter them according to our aims. The main goal of this thesis is to develop a computational model that is capable to classify drug-protein pairs concerning binding affinity, in a target-oriented form. For some specific proteins, there are several compounds that have experimentally tested for target inhibition. That central target, though, according to cell signaling pathways analysis, is the exact previous protein in the pathway from the protein aiming to inhibit. For example (see Figure 5) to inhibit ERK 1/2 it needs to measure binding affinity at MEK 1/2.

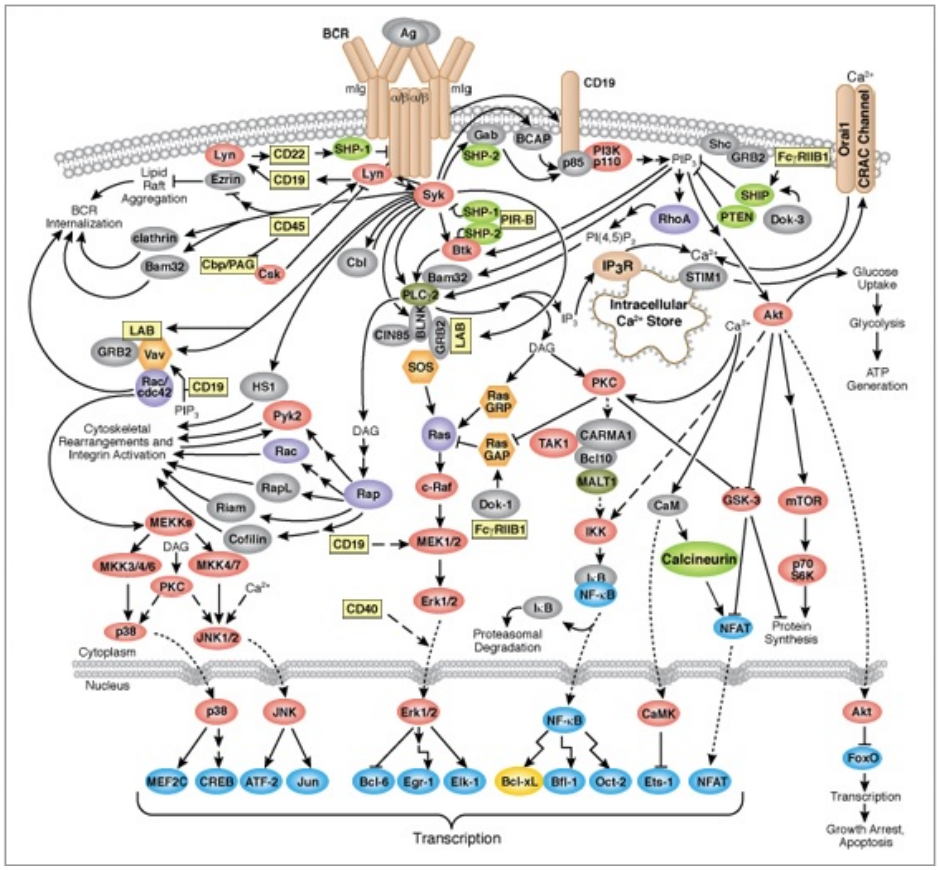

Figure 5: Cell signaling pathways.

Therefore, specific targets have to be selected for later study, depending on their drug discovery importance and the bioactivity, which is used for this thesis' purposes as has already discussed, IC_{50} value. At Table 1 the final selection of proteins with the most samples for IC_{50} is presented.

Table 1: Selected proteins for preprocessing

Targets	Number of compounds
P38	6128
PI3K	4807
AKT-1	3236
JAK-1	2543
ERK-2	2475
JNK	1706

2.2.2 Data filtering and normalization

After obtaining all the compounds data for each protein, noise filtering is required. In the beginning, the first steps of filtering are to remove samples without IC_{50} values and also samples without specific experimental assay type, because it is important to know if the trial took place at a binding or not binding assay, and then split them depending on those two categories in order to normalize the data with different thresholds later.

After editing and fixing IC_{50} values, normalization was performed according to the equation:

$$IC_{50_{norm}} = -log_{10}(IC_{50}/10^9). \tag{2.1}$$

Moreover, each compound of the samples is named by a different id from ChEMBL which is used to import the simplified molecular-input line-entry system or else canonical smiles, which is a string representation for compounds to describe the chemical structure. Those strings are imported from an open database called ChEMBL (Mendez et al., 2018). However, not all of the compounds are located in that database leading to import them manually from other sources. Unfortunately, that difficulty leads to remove some of our targets due to the lack of samples with known canonical smiles.

Regarding the classification output, meaning whether a drug binds with protein or

not, drugs were classified as binders if the normalized IC_{50} value was above 7 and 6.5 for data from binding and non-binding assays respectively. Finally, we chose to keep three of them with the most samples which are P38, PI3K, and AKT-1 (see Table 2).

Table 2: Final targets for learning

Targets	Number of compounds
P38	3559
PI3K	3754
AKT-1	2140

2.3 Dataset splitting

In Deep Learning, evaluating the performance and robustness of the models is of utmost importance. On this front, it is necessary to appropriately split the data into multiple training and validation sets. In this study, 6-fold cross-validation is chosen to split the data, with respect to diversity between train and validation data for more accurate results. Specifically, the splitting of the data is based on the following strategy:

- Start with a molecule which is the one with minimum average distance with all the others.
- Grab the molecule which is closest to it based on Tanimoto similarity of the extended connectivity fingerprints and put these in the same fold.
- Continue and grab the next molecule which is the closest to them (closest to either of those) and add it.
- Stop when the fold is complete

Finally, six different folds of data for afterward training for each of the three specific targets, were created, plus an extra set for testing with more difficult splitting was reserved for later evaluation. At tables 3, 4 and 5 the number of compounds per set are presented.

Table 3: Cross-validation folds - P38 target

Folds	Training compounds	Validation compounds
1	2541	509
2	2541	509
3	2541	509
4	2541	509
5	2541	509
6	2545	505
Test	3050	509

Table 4: Cross-validation folds - PI3K target

Folds	Training compounds	Validation compounds
1	2680	537
2	2680	537
3	2680	537
4	2680	537
5	2680	537
6	2680	532
Test	3217	537

Table 5: Cross-validation folds - AKT-1 target

Folds	Training compounds	Validation compounds
1	1528	306
2	1528	306
3	1528	306
4	1528	306
5	1528	306
6	1530	304
Test	1834	306

Proteins' datasets were also split into two more sets, one with random split method and one with ave bias method. These extra sets were generated in order to discuss the performance's variation among the different data sets and if it depends on the compounds' similarity of the training and validation data. For the random sets, data was split 85% training and 15% validation, from the whole dataset for every target, using the train-test splitting function from python package Scikit Learn (Pedregosa et al., 2011). The last splitting method, called Asymmetric Validation Embedding (AVE) Bias, used only with data from P38 to discuss models' performance with very difficult to predict compounds. AVE is a measure of training-validation redundancy for ligand-based classification problems, that accounts for the similarity among inactive molecules as well as active ones (Wallach & Heifets, 2018).

3 | Material and methods

The main goal of this thesis is to implement a Deep Learning model to predict the binding affinity of a compound with a specific target/protein. At first, because the binding problem is one of the most important tasks at computational drug discovery, it is wise to make research on previous methods, as discussed at Section 1.4, and then test their results at our data. Apparently, results were not significantly different from the original because those models were not able to produce better performance. Therefore,in this chapter will be discussed suggestions of classification techniques on the binding affinity topic.

3.1 Molecular Features

Before analyze the proposed methods, it is necessary to describe the input data of the upcoming machine learning models. As reported at Chapter 2, canonical smiles are the string representation of the compounds corresponding to their chemical structure but in order to import the data into the model, they need to be transformed into other representations. In this study, we used two different types, extended connectivity fingerprint and molecular graphs.

3.1.1 Extended connectivity fingerprint

ECFPs are circular topological fingerprints designed for molecular characterization, similarity searching, and structure-activity modeling (Rogers & Hahn, 2010). They are among the most popular similarity search tools in drug discovery and they are effectively used in a wide variety of applications.

The main features of ECFPs are the following:

- They represent molecular structures by means of circular atom neighborhoods.
- They can be very rapidly calculated.
- Their features represent the presence of particular substructures.
- They are not predefined and can represent a huge number of different molecular features (including stereochemical information).
- They are designed to represent both the presence and the absence of functionality, since both are crucial for analyzing molecular activity.
- Their generation method (Fig. 6) can be flexibly customized to produce various types of circular fingerprints for diverse applications.

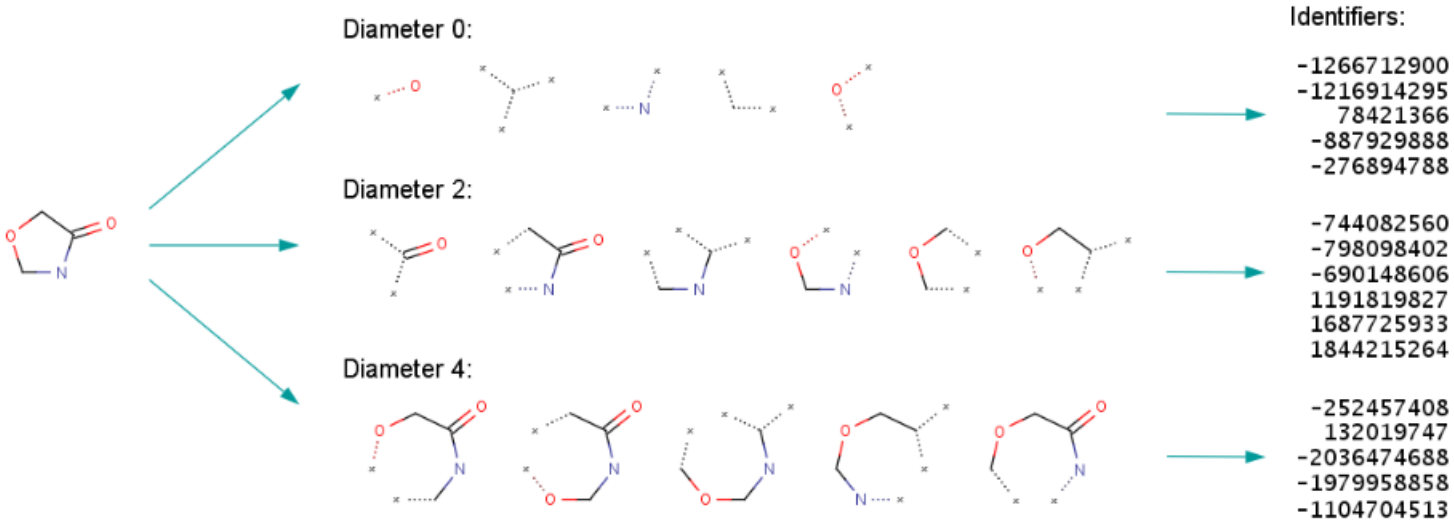

Figure 6: ECFP generation process

3.1.2 Molecular graphs

Molecular graphs are presented to the model using the Atom array, the Edge array and the Bond array. The Atom array has as many rows as the max number of atoms across all compounds and each column represents an atom feature. In total, 62 atom features are utilized. The atom features consist of the concatenated vectors of 4 one hot encoded features and 1 binary feature, which describe:

- The symbol of the atom (one-hot).
- The degree of the atom (one-hot).
- The number of attached hydrogen atoms (one-hot).
- The valence of the atom (one-hot).
- If the atom is aromatic (binary).

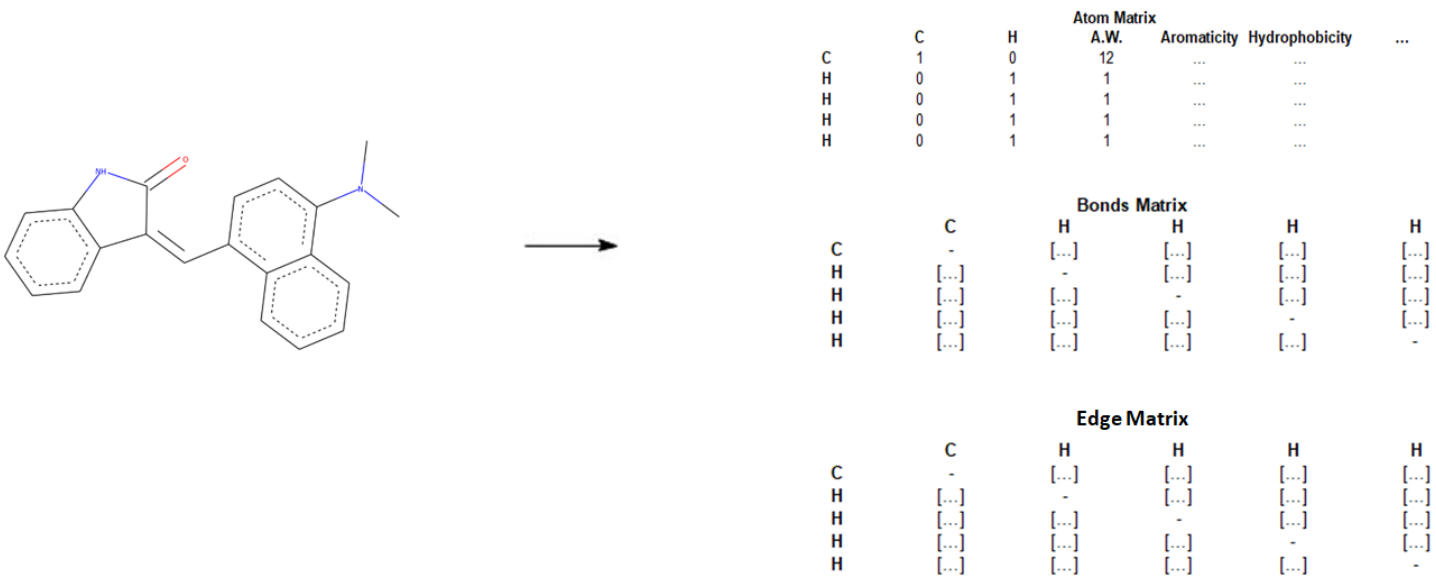

Figure 7: Compounds as graphs

The Edge array describes the connectivity of the graph representing the molecule.

The Edge array consists of as many rows as the max number of atoms. Each row contains the atom's neighbors. The Bond array is 3-dimensional and contains the features of each bond. Each row represents an atom, while each column represents a neighbor, up to 5 for each atom. A bond is described by 6 binary features contained in the Bond array, which describe whether the bond is:

- Single
- Double
- Triple
- Aromatic
- Conjugated
- In a Ring

The Atom, Bond and Edge arrays were created using RDKit ("Open-source cheminformatics", n.d.) in python.

3.2 Evaluation metrics

The desired metrics for model evaluation were chosen concerning the classification problem. All the models were trained according to receiver operating characteristic curve, or ROC curve, with area under ROC curve as a metric which describes the training process, meaning that the training is perfect if the AUC score is equal to one. Moreover, for the predicted values, the scores that were calculated, are the mean average precision (MAP), precision, recall, and accuracy, using the confusion matrix from the comparison of predicted and real values.

3.3 Optimization

To improve models' performance, we did hyper parameter optimization using the python package Hyper-opt (Bergstra, Yamins, & Cox, 2013). This optimization approach consists of four components:

- **Null distribution specification language** that is about an expression language for specifying the hyperparameters of a search space. This language describes the distributions that would be used for random, unoptimized search of the configuration space, and encodes the bounds and legal values for any other search procedure. A null prior distribution for a search problem is an expression G written in this specification language, from which sample configurations can be drawn.
- **Loss Function** which is the criterion we desire to minimize and maps legal configurations samples from G to a real value. The desired loss function in this study is average precision score.
- **Hyperparameter Optimization Algorithm (HOA)** which takes as inputs the null prior distribution G and an experimental history H of values of the loss function, and returns suggestions for which configuration to try next.
- **Database**, at which is stored the experimental history H of configurations, that have been tried, and the value of the loss function at each one.

3.4 Baseline models

We implemented some baseline models to compare with our suggested techniques. The first one is a simple deep neural network with three hidden layers, fully connected with input and output the canonical smiles and predicted label respectively.

Moreover, other machine learning algorithms such as random forest, k-nearest neighbors, support vector classifier, and logistic regression are tested for comparison with our models.

3.5 XGBoost model

From machine learning methods, the best performing is the XGBoost. XGBoost stands for eXtreme Gradient Boosting and it is one of the newest and most used machine learning techniques nowadays. It was preferred to test that method despite not being a traditional neural network because of the many capabilities it has. At first, one of its features is the gradient boosting algorithm that gives better performance and faster training than other similar methods, such as random forest, based on decision trees. Besides, that model has as input the molecular fingerprints of canonical smiles at ECFP4 format and as output the classification label of binding.

XGBoost model was also optimized depending on protein's P38 data for better results with Hyper-Opt package in Python. At Table 6 the final training parameters of the model are presented.

Table 6: Training parameters of XGBoost

Max depth	8
Learning rate	0.40236
Number of training rounds	300
Booster	Gradient oosting Tree
Objective function	Binary Logistic
Evaluation metric	Auc

3.6 Deep GCN model

Deep graph convolutional network (Deep GCN) is the first deep learning model and is identified as the base model of this thesis.

In the beginning, the inputs of the model are molecular graphs, as discussed at 3.1.2, because with that method the model is able to learn and understand better chemical structures. After that, these graphs pass through a graph convolution neural network.

Graph convolutions were implemented in Keras (Duvenaud et al., 2015). A graph convolutional layer aggregates information from the neighboring nodes of a node-atom in the molecular graph. For every atom, its bond features are summed and concatenated with its atom feature vector. The resulting feature vector of each atom is summed with the feature vectors of its neighbors, using the connectivity information of the Edge array, creating in this way a new feature vector for every atom with aggregated information from the atom's neighborhood. Then, every feature vector passes through a fully connected layer, based on the atom's degree, and a non-linear activation function. Typically, following a graph convolution layer, a function, such as sum, is used to aggregate node embeddings into whole graph embeddings. In our implementation we omitted the use of an aggregation function and instead utilized 1D convolutions to gather information across neighborhoods and produce a graph feature map.

1: **Input:** Atom array X_A, Bond array X_B, Edge array D

2: **for** each atom a_i in a molecule

3: $SX_{B_i}=\sum X_{B_i}$

4: $X'_{A_i} = concatenate(X_{A_i}, SX_{B_i})$

5: **for** each neighbor j from N neighbors

6: $SX_{B_j}=\sum X_{B_j}$

7: $X'_{A_j} = concatenate(X_{A_j}, SX_{B_j})$

8: $X''_{A_i} = X'_{A_i} + \sum_{j=1}^{N} X'_{A_j}$

9: $X_{A_i}^{new} = relu(W_{degree} * X''_{A_i} + b_{degree})$ #is the new concatenated atom and bond matrix

Figure 8: Graph convolutional layer pseudocode

Moreover, after the encoder, there is a two-layer fully connected neural network that has the encoder's output as input and then predicts the label of the drug that means if it binds with the target or not.

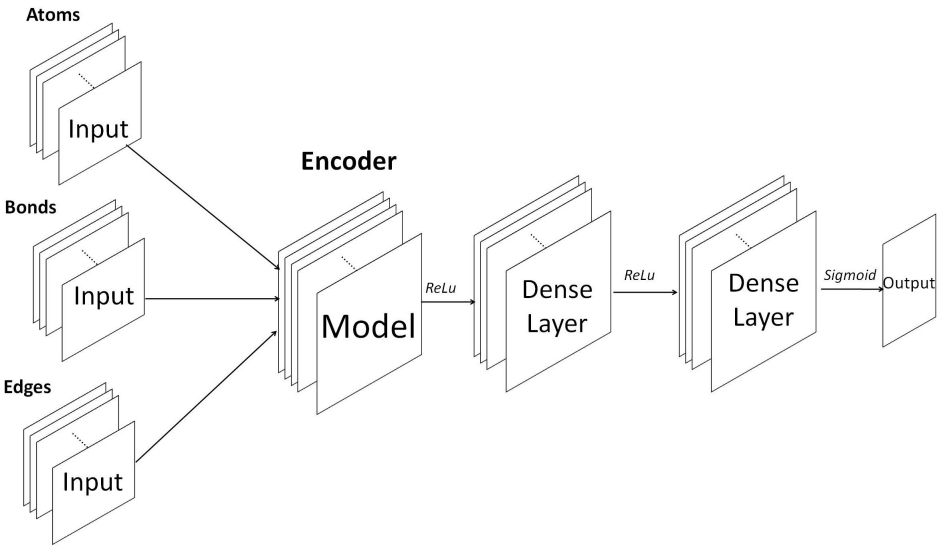

Figure 9: Deep GCN model

Finally, Deep GCN is trained regarding binary cross-entropy loss function that is given from equation 3.1, where y is the label, 1 for binding and 0 for not binding, and p(y) is the predicted probability of the drug's binding to be positive for all N drugs.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i)) \tag{3.1}$$

That model has also be optimized using the python package hyper opt. The optimization took place according to the average precision score of 6-fold cross-validation on P38 data. Table 7 presents the hyper-parameters of the model.

Table 7: Hyper-parameters of GCN

Optimizer	Adam
Learning Rate	0.007037
Epochs	35
Batch Size	64
No. Dropouts	2
Activation function - Fully connected	ReLu
Activation function - Output	Sigmoid

3.7 One-shot learning

The third method used in the study of binding is the one-shot technique (Altae-Tran, Ramsundar, Pappu, & Pande, 2017). Generally, this technique is chosen when the number of available data is small, just like in the present issue, whose number does not exceed two and a half thousand, giving though improved results compared to a simple neural network. In particular, the basic idea of a one-shot model is the existence of two or more similar in relation to architecture neural networks, called siamese networks, from which their output is compared according

to a cost function. The output, in this case, is a mapping of the three characteristics of the drugs in a vector that is called embedding, and the cost function, that has chosen for the training, is the triplet loss.

That loss function is calculated according to three features: the anchor, the positive, and the negative. The positive has the same label as the anchor with the negative to have the opposite for binary classification and all three together are a triplet of data. Every triplet that comes through the siamese neural network ends up to an embedding vector which is split to anchor, positive and negative embeddings. After that, it is needed to compute the pairwise distances of them and especially the distance of the anchor from the positive and the distance of the anchor from the negative and finally calculate the triplet loss value through equation 3.2.

$$L_{triplet} = max(d(a, p) - d(a, n) + margin, 0.0) \tag{3.2}$$

That addition of margin at the loss affects the training of the model because a bigger margin means easier triplets while smaller or zero is connected to harder to learn triplets. During minimization of the loss, the distance of the anchor to the positive becomes smaller than it with the negative leading to better classification of the data.

The main advantage of the one-shot method is the capability of learning with high performance with a few data. That is explained by the triplets, because for example from k data points may be able to construct k^3 triplets at most without check for validity though. The problem, in that case, is how to construct these triplets. There are two known algorithms of triplets mining, the off-line and online mining.

3.7.1 Offline triplet mining

In offline triplet mining, the triplets are created before train the model at another pre-processing algorithm. The only need for this method is to identify valid triplets, with anchor and positive with label 1 and negative with 0 despite being hard or easy triplets. The quality of them is depending only on the pre-processing algorithm which concludes to be a more time-consuming method with not so much promising results.

In particular, the off-line mining model consists of three siamese graph convolutional networks, the same as the simple GCN, one for every of the anchor, positive and negative triplet's compounds. The inputs, in this case, are also the atoms, the bonds, and the edges of the drug for all the models respectively. However, their outputs are the embeddings of these drugs to merge them and then calculate triplet loss to train the model as shown at Figure 10.

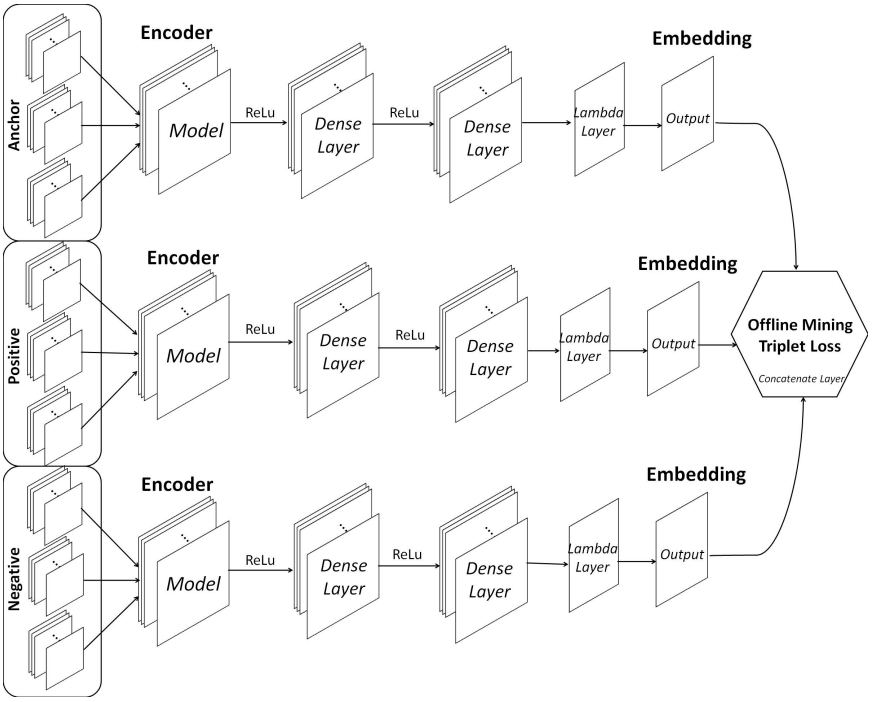

Figure 10: One-shot offline triplet mining model

Furthermore, the triplets have been constructed by hand depending on training difficulty at three categories, the easy, the hard, and the random triplets. Easy triplets conclude anchors that have smaller distances from positives than negatives, hard triplets contain compounds with the opposite feature and the last ones are completely random but valid triplets. The last category was built only to test the model at too many data points. In the end, to evaluate the model, it was used an XGBoost model taking as input the predicted embedding and then was asked to predict the label of the drug. Training parameters for both neural network and xgboost are the same from Table 3.1 and Table 3.2 correspondingly with the addition of the margin that was set to 0.5.

3.7.2 Online triplet mining

In online triplet mining, triplets are generated during the training with the help of a different but similar version of triplet loss. The main concept of this method is that for a batch of data B it can generate more triplets than the off-line one with respect to triplet validity. Data generation is happening through the loss to succeed better loss minimization, performance, and accuracy.

The on-line triplet mining model is based on that loss and is more simple than the previous one. Instead of the siamese networks, there is only one model, with input the drug's graph representation and its label and output its embedding (see Figure 3.3). The difference is at the extra input of the labels that are needed to generate the triplets because modified triplet loss is using them to split data to anchor, positive, and negative and then finally build valid triplets. Evaluation in this case was done also by using an XGBoost model using the embedding.

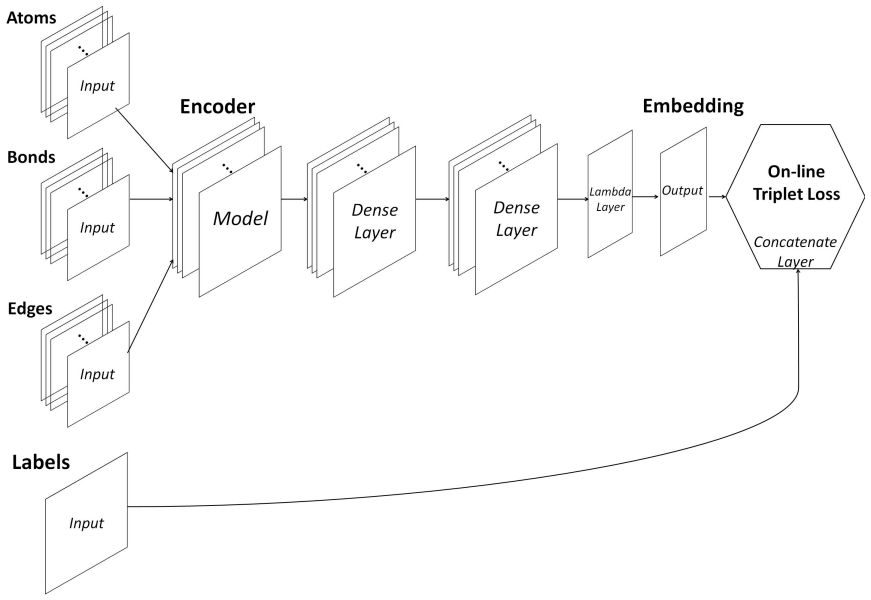

Figure 11: One-shot online triplet mining model

Because of these methods better performance compared to other implementations, it was utilized in combination of the xgboost method, and their hyper-parameters were optimized simultaneously. These hyper-parameters are demonstrated at Tables 8 and 9.

Table 8: Hyper-parameters of on-line triplet mining model

Optimizer	Adam
Learning rate	0.000811
Epochs	25
Batch size	288
No. Dropouts	2
Activation function - Fully connected	ReLu

Table 9: Training parameters of XGBoost - Optimized for embeddings

Max depth	7
Learning rate	0.41409
Number of training rounds	300
Booster	Gradient Boosting Tree
Objective function	Binary Logistic
Evaluation metric	Auc

3.8 Multitask learning

The next and last method that will be presented is multitask learning. It is a new technique (Ramsundar et al., 2015) based on a model that is capable of learning multiple tasks with the same data. In this case, it will be used to predict only one task, the binding affinity, but for all of the three proteins, meaning that the input data will increase, and the neural network will be trained on all of the compounds and after that will predict separately the binding affinity. So, this method will be tested with two network architectures, the graph convolutional network from section 3.1 and the on-line triplet mining model from 3.3.2.

3.8.1 Multitask GCN

Regarding, the GCN architecture, the differences are located at the inputs and the outputs of the model. More specifically, there are nine inputs, one triplet of atoms, bonds, and edges for each protein's drug which passes through a different encoder. Also, the three outputs of the encoders are combined as one input for the fully connected neural network and before output, the tensor is split into three parts, one for every protein, to input at the final output layer. That network layer is responsible for predicting the label of each drug. Finally, the training parameters are presented at table 10.

Table 10: Hyper-parameters of multitask GCN model

Optimizer	Adam
Learning rate	0.0007
Epochs	30
Batch size	256
No. Dropouts	2
Activation function - Fully connected	ReLu

3.8.2 Multitask on-line triplet mining

The second and last use of the multitask learning method is on Section 3.3.2, which is related the on-line triplet mining. The modification is like the previous multitask method, with the same nine inputs but this time with different outputs and with the addition of three extra inputs, the labels of each protein's drugs. The outputs are the embeddings of the drugs and not the predicted label because of the triplet loss function. Finally, the predicted embeddings are evaluated with the help of three xgboost models, one for every target.

4 | Results

The implementations of machine learning methods, that were used to achieve the goal of predicting binding affinity, were discussed in Chapter 3. In this chapter, the results of training them on the different types of splitting sets will be presented.

4.1 Performance evaluation for P38

Target P38 were evaluated according to three different training-validation sets: the test set, the random split set, and the AVE set. Test set used to train and predict values for all of the models, including baselines, random set tested on baseline models, and all suggested methods apart from the offline triplet mining, and finally the ave set on GCN, XGBoost and online triplet mining.
At figures 12 and 13 are demonstrated the performance on test and random set for the P38 dataset on baselines respectively.

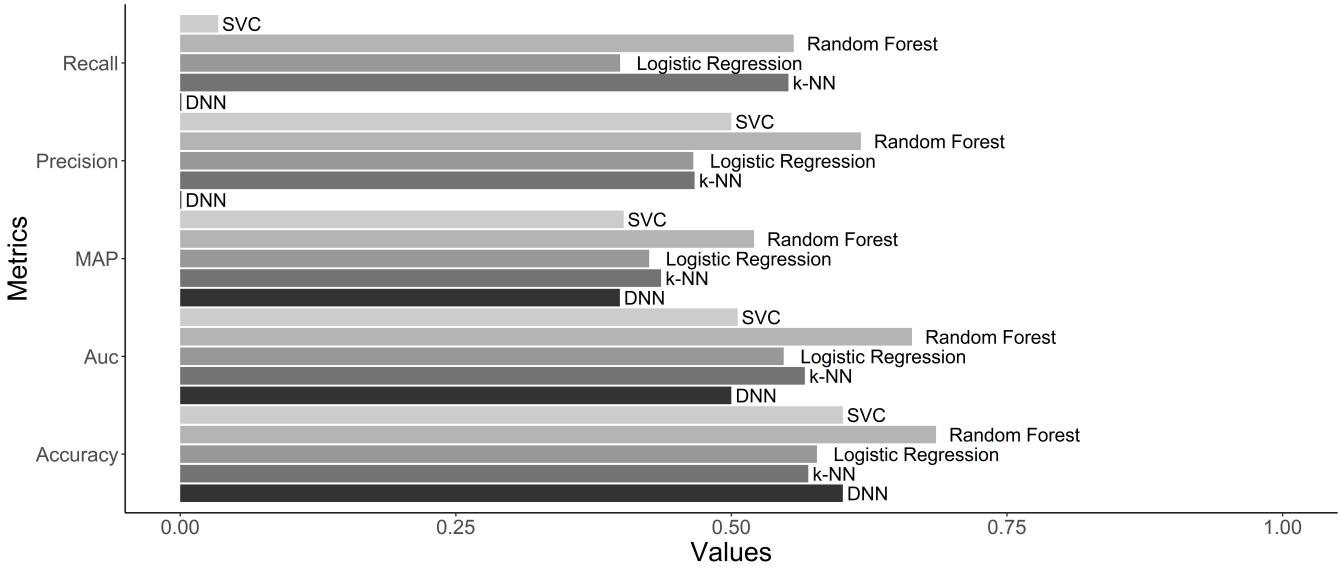

Figure 12: P38 - Baselines on test set

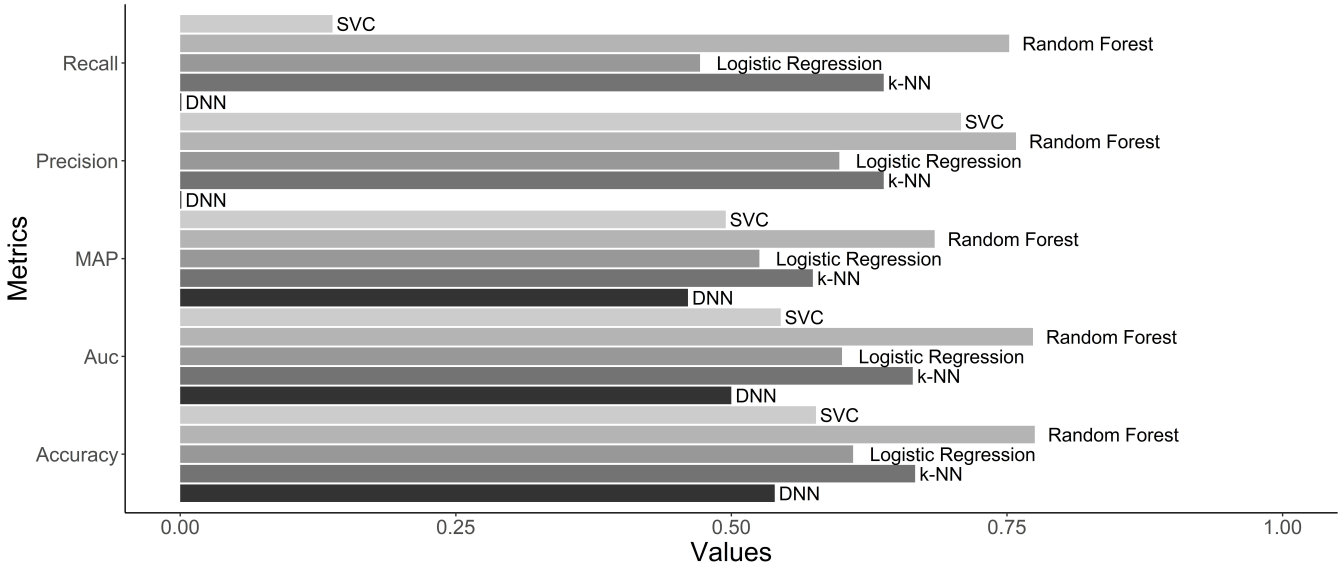

Figure 13: P38 - Baselines on random set

As expected, baselines' performance was reasonably not very good, especially on test set, with the best precision to be around 62% for the random forest. Also, at random set all the scores were better comparing to the test set.

After the traditional machine learning methods, P38 were evaluated on our proposed deep learning techniques for binding affinity prediction. The results from the three different kinds of training-validation sets are presented at the next figures.

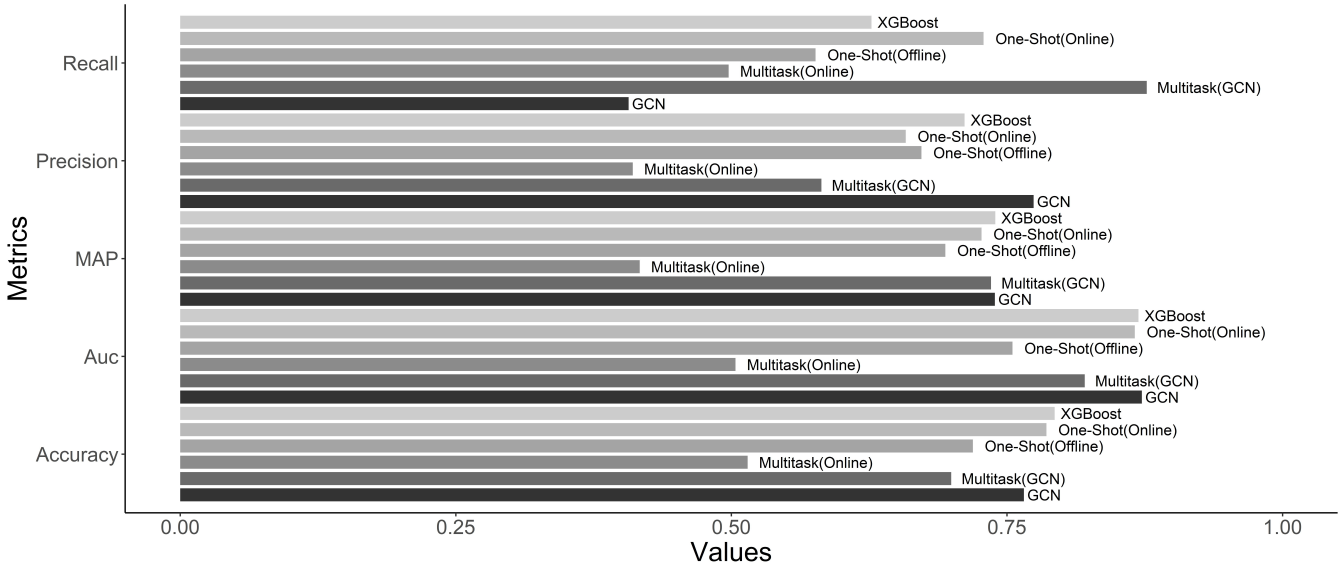

Figure 14: P38 - Performance evaluation on test set

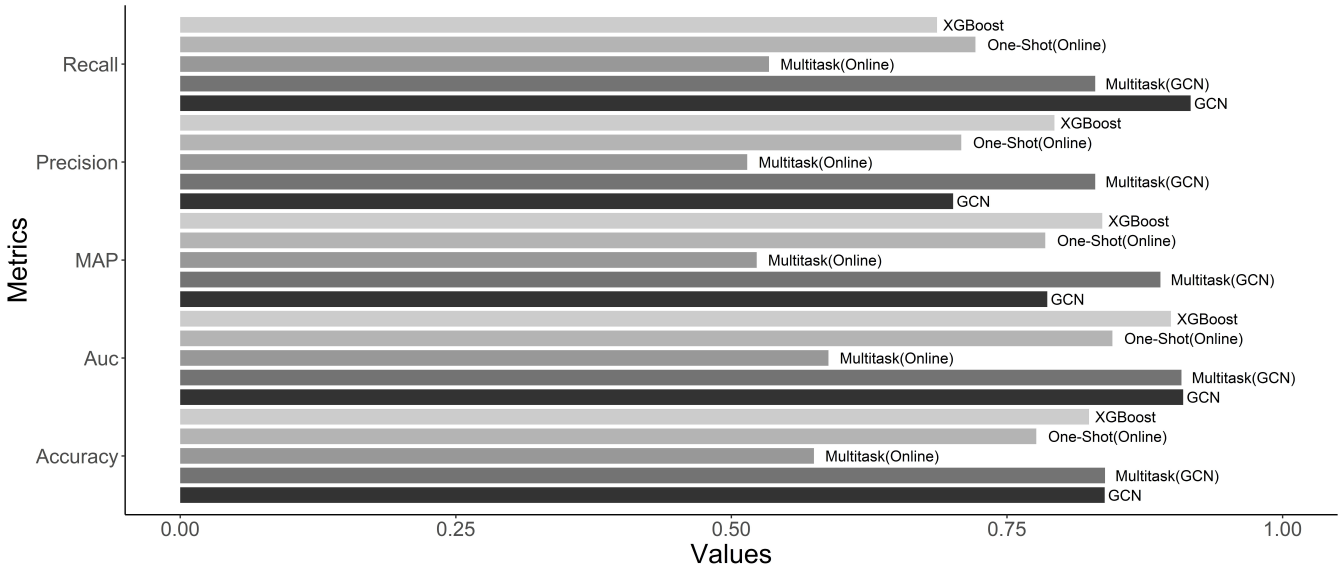

Figure 15: P38 - Performance evaluation on random set

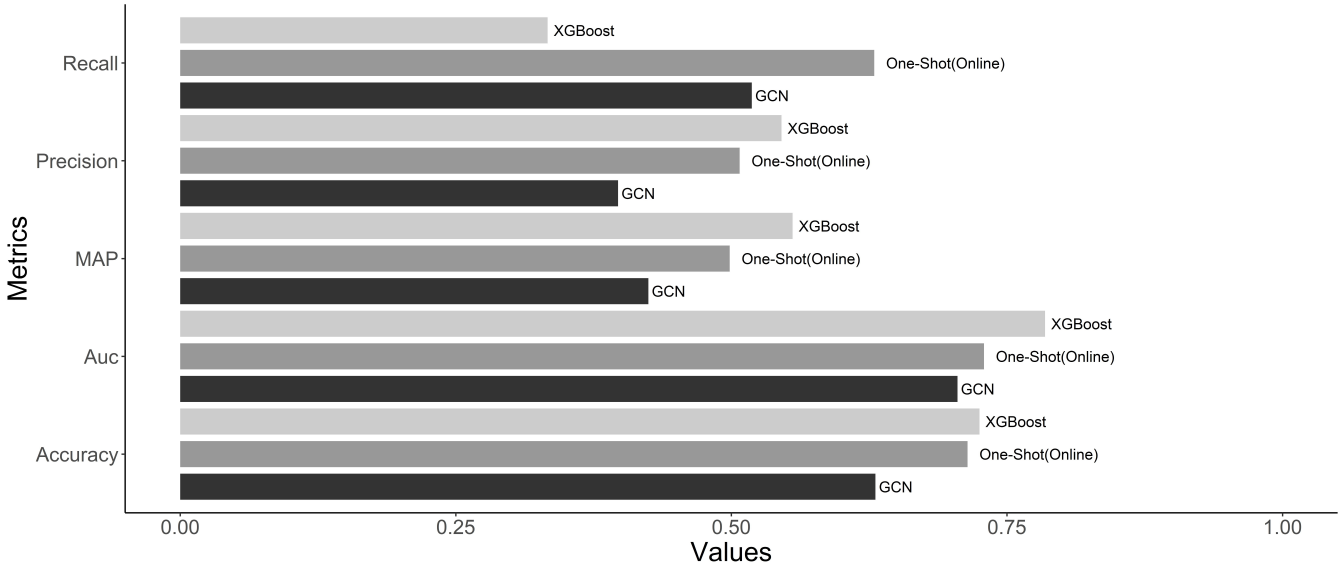

Figure 16: P38 - Performance evaluation on AVE Bias set

The first observation among the figures, for the same type of models, is the relation of the performance with the set's difficulty. Harder splitting leads to worse efficiency on predicting accurately the label of a compound from the validation data, due to small similarity with training's data compounds. Moreover, regarding the results on test set, xgboost performance is the most stable with less variations between the metrics' values apart from the recall. However, xgboost's training parameters were optimized for that target so these results would not be better. For deep learning models, GCN, that was also optimized, has the best map score and precision with 74% and 76% respectively and also high accuracy. Finally, online triplet mining model has almost the same map with the gcn and the besy recall. One-shot model, with offline triplet loss, and multitask with online mining, seems to be the worse.

4.2 Performance evaluation for PI3K

The second protein's data, were split at two kinds of sets, test and random. Both of them have been tested on baseline models and our methods, aside from the one-shot with the offline triplet loss function for the random split set. At figures 17 and 18 the baselines' results are shown for PI3K.

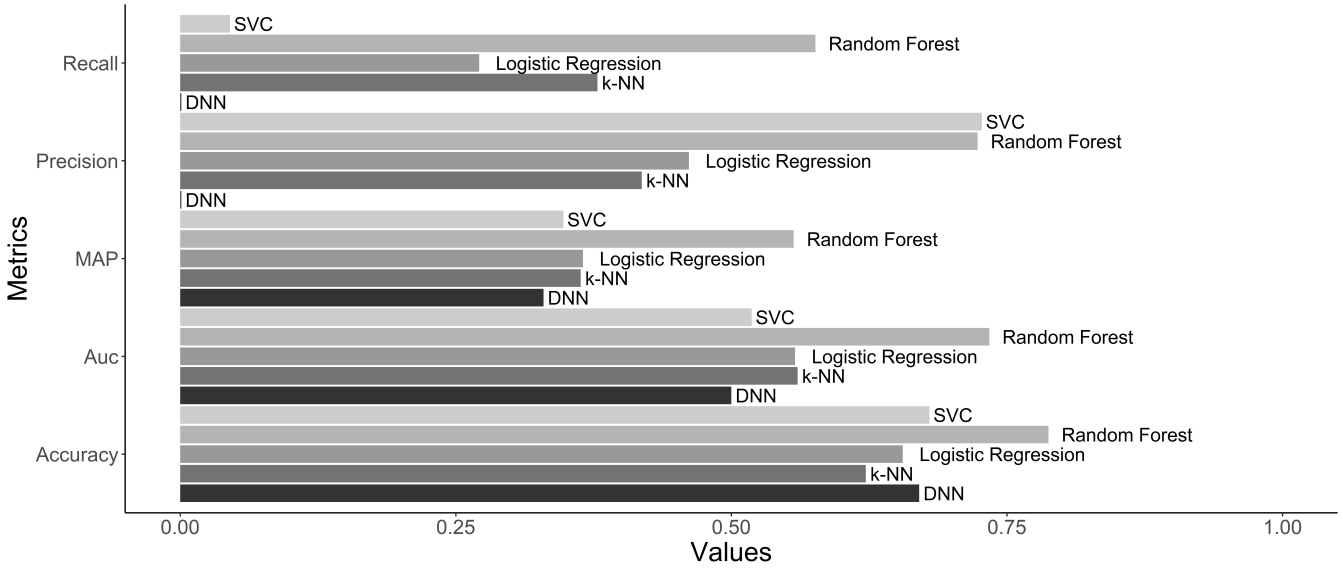

Figure 17: PI3K - Baselines on test set

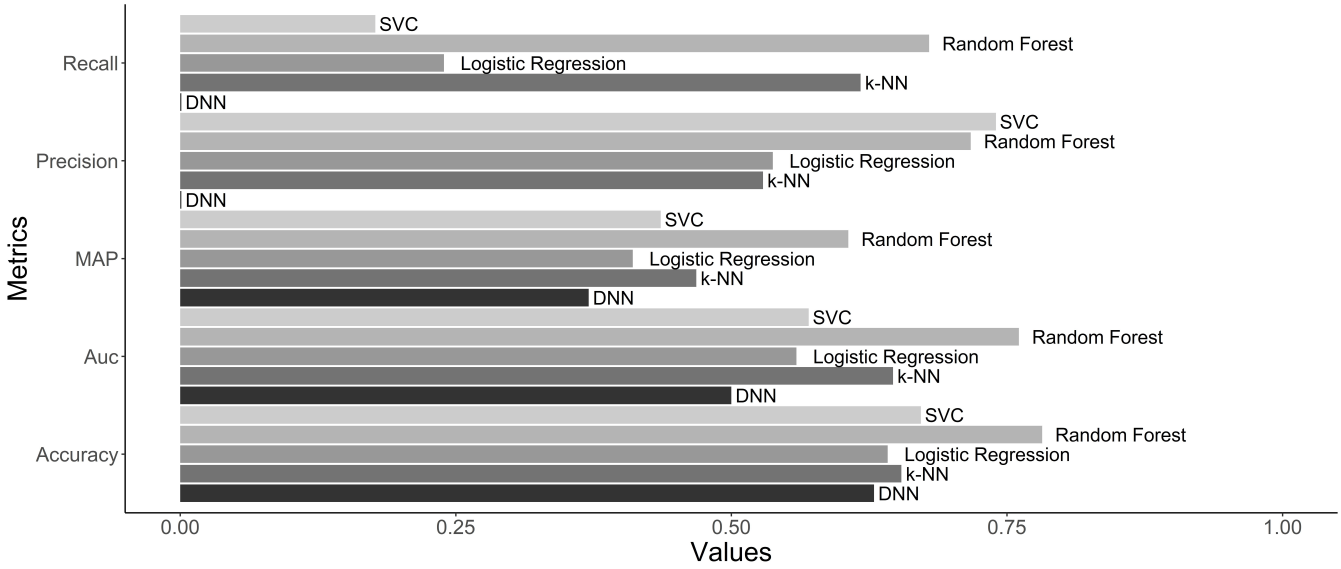

Figure 18: PI3K - Baselines on random set

Just like P38, random forest is the best performing model from the baselines at this target among the others, with only the support vector classifier having the best precision score but also bad overall performance.

At next, our methods tested on PI3K's sets as presented on the upcoming figures.

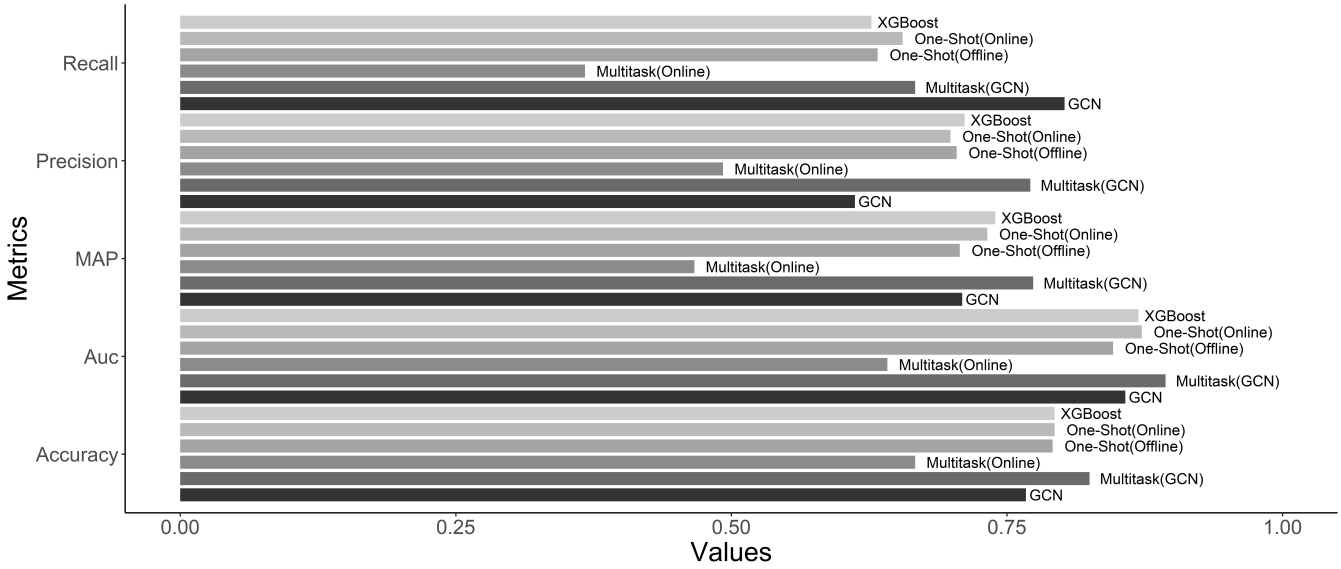

Figure 19: PI3K - Performance evaluation on test set

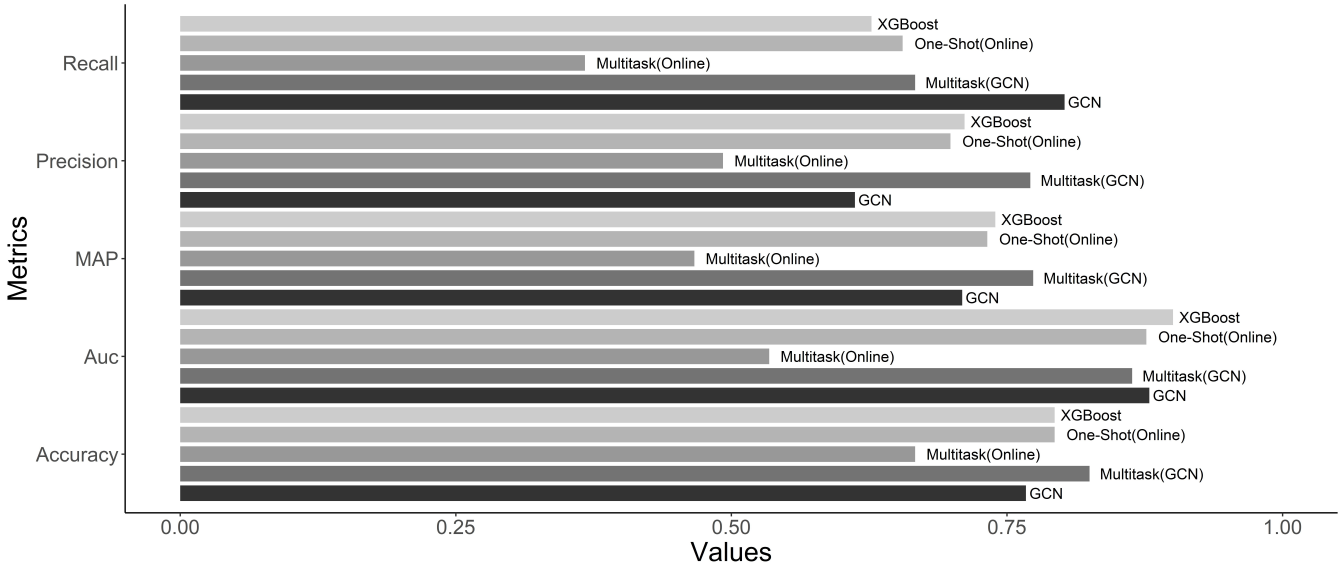

Figure 20: PI3K - Performance evaluation on random set

In this case, multitask learning, based on GCN, model has the best performance between all the other techniques, apart from the its recall, which is not the best overall. Map and precision scores are above 75% and accuracy is almost 80%

leading to a more generalized model, capable of learning features from the compounds and predict binding affinity more efficiently. Also, at this target, the performance-splitting relation is not very clear, because their evaluation results are very close.

4.3 Performance evaluation for AKT-1

The last target, AKT-1, was tested, as PI3K, with two types of sets at the same models.

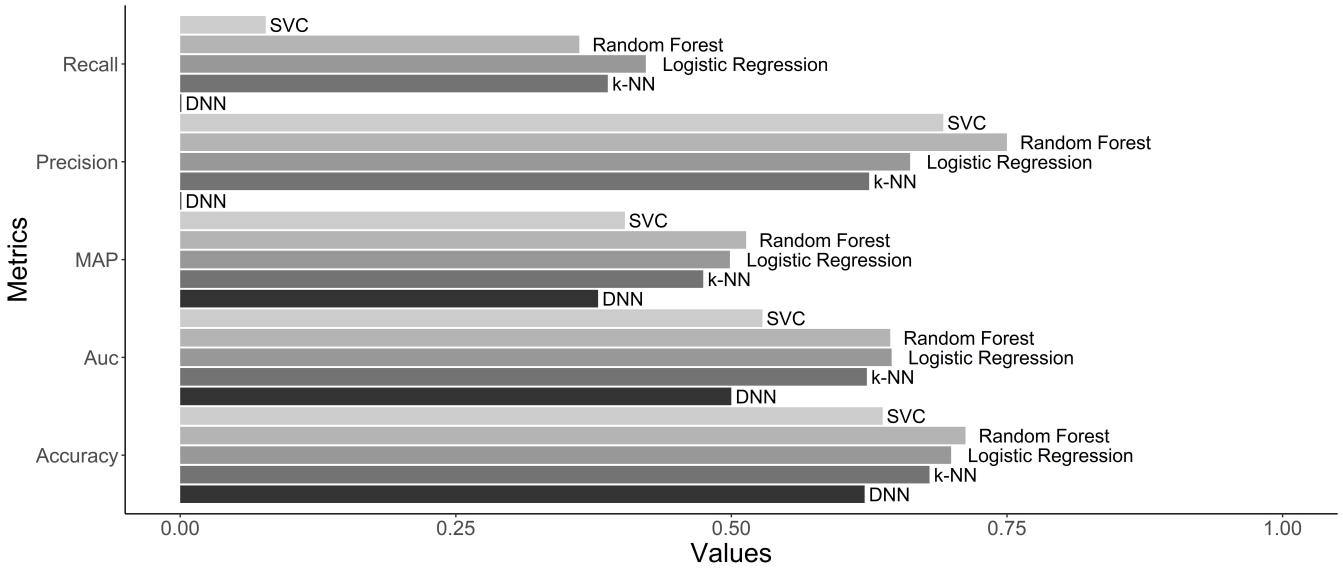

Figure 21: AKT-1 - Baselines on test set

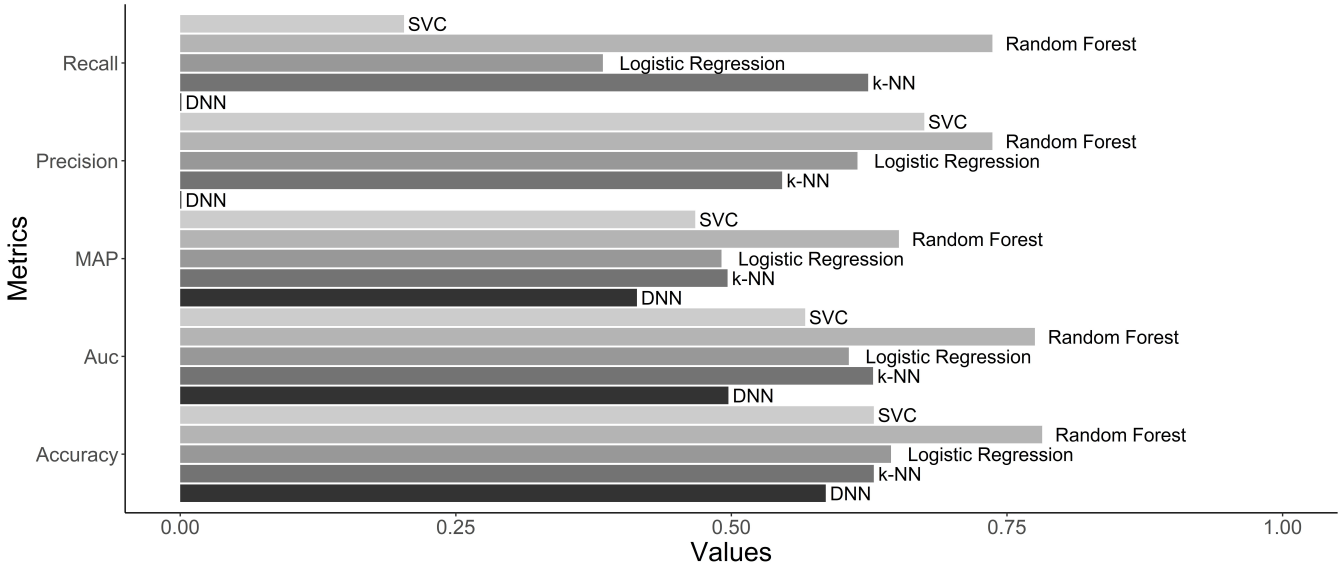

Figure 22: AKT-1 - Baselines on random set

To begin with the baselines, there are also two more algorithms, k-NN and logistic regression which are performing at the same level with random forest. On the other hand, at the random set only random forest is much better among all the baselines.

Eventually, for our methods, the results for AKT-1 are presented at figures 23 and 24. At first, between test and random set, there is not significant difference about the scores which is possibly related with the small number of data points and better training and feature extraction from the molecules. Also, it is not able to choose one as the best model because all of them are concluded to the same scores at our metrics, except recall, with the GCN to have a small increase, comparing to the others, at the test set.

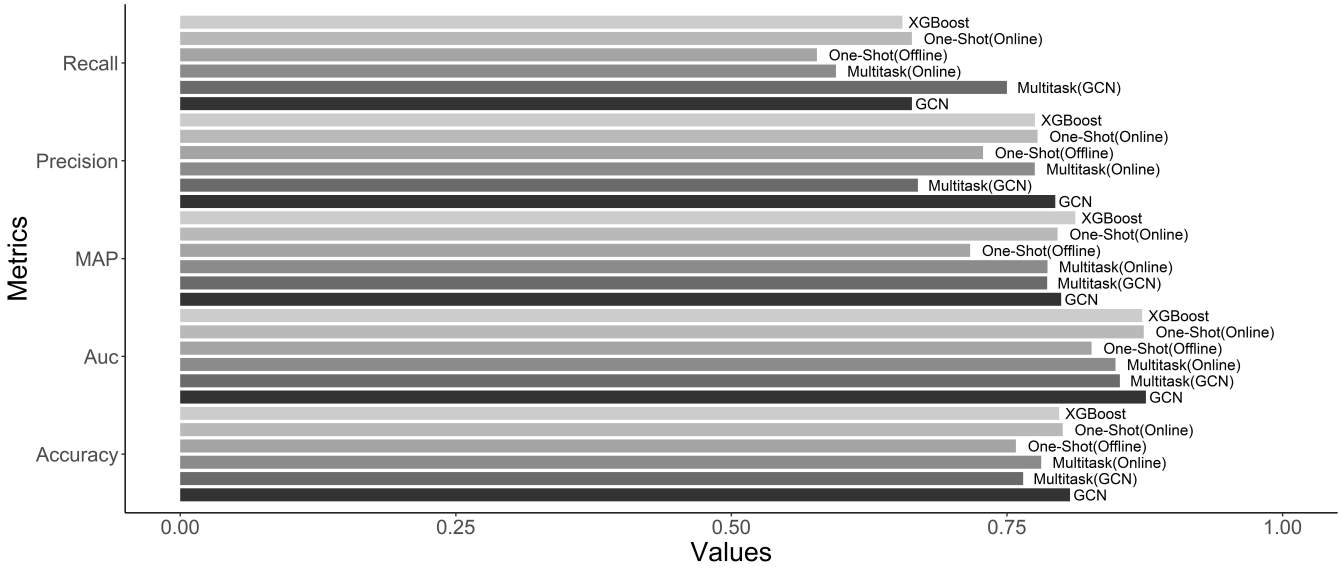

Figure 23: AKT-1 - Performance evaluation on test set

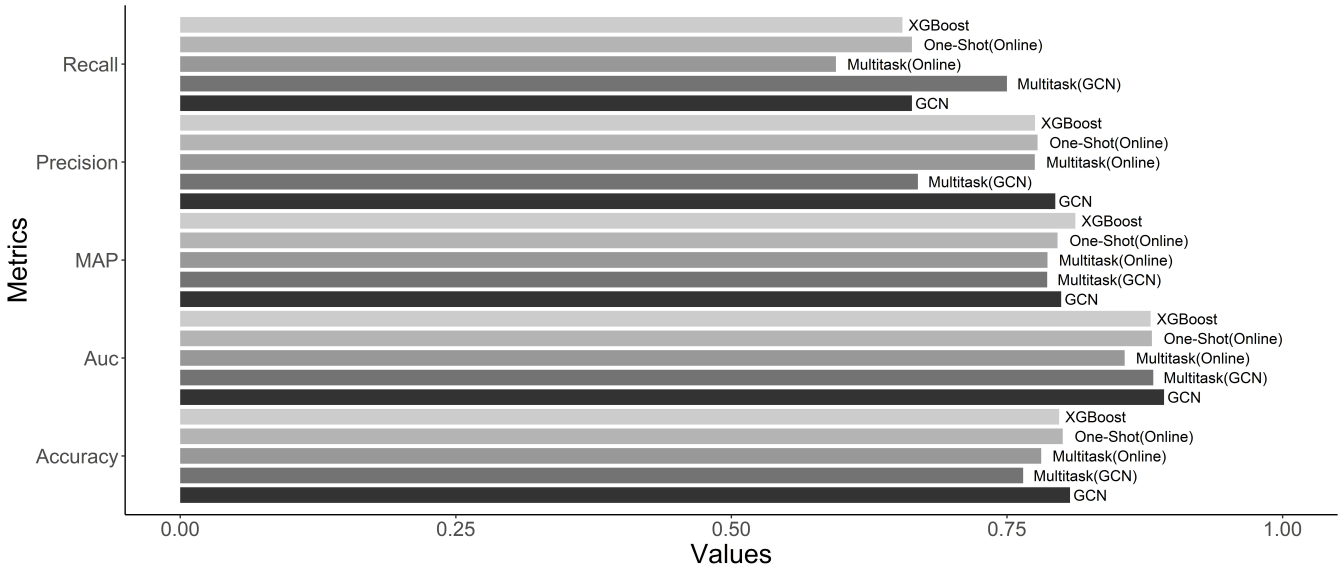

Figure 24: AKT-1 - Performance evaluation on random set

5 | Discussion and conclusion

Computational drug discovery is definitely a huge breakthrough in drug design, helping to test compounds efficiently faster than other methods, as HTS, at the first stages of design process.

In this study, the aim was to implement a computational model using machine learning methods to predict the binding strength at a drug-target pair. The problem with this approach was the type of data, due to lack of 3D structures in them, leading to use the encoder to transform smiles to molecular graphs. Binding affinity corresponds to proteins' binding sites which is difficult to find and represent only with the encoder and the neural network and is needed to take advantage of protein's and drug's 3D features.

On the other hand, artificial intelligence overpasses that kind of problems in drug discovery, with multiple of techniques, capable of understanding and learning biological and chemical features from the data. More data means better performance but with no restrictions for methods that give better results with a few and better quality data points, such as one-shot models.

The most important task in this study was to make a generalized model for all of the kinds of datasets which was not able to accomplish due to lack of computational power and data. For example with the ave bias set, the performance scores were significantly lower than the test set and random split set, which verifies the importance of the similarities among drugs from the training and the validation dataset. Nevertheless, the variation among the other two sets was not so major meaning that these models are more robust to the split.

Our methods performed pretty well concerning the challenging binding problem and the difficult splits especially on the test set. Moreover, the most important metric in our case is the mean average precision, because this score depends on

the number of false positives drugs, which means less false positively binders will pass through this stage of drug design for experimental trials. MAP is more robust and balanced than the simple precision score because of being calculated across all of the classes' precision.

In conclusion, GCN, XGBoost and Multitask GCN seems to be the best models in this study and for these three specific targets. As discussed previously, they had the best map score for all the three targets at the test set leading to be satisfactory computational models for the binding topic and drug design. Of course they can be improved and especially the multitask model, with further targets or tasks to increase the number of data, more complex neural networks and additional techniques, to predict, in the end, more accurate the binding affinity.

Bibliography

Altae-Tran, H., Ramsundar, B., Pappu, A. S., & Pande, V. (2017). Low data drug discovery with one-shot learning. *ACS Central Science*, *3*(4), 283-293. (PMID: 28470045) doi: 10.1021/acscentsci.6b00367

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th international conference on international conference on machine learning - volume 28* (p. I–115–I–123). JMLR.org.

Cichonska, A., Ravikumar, B., Allaway, R. J., Park, S., Wan, F., Isayev, O., ... Aittokallio, T. (2020). Crowdsourced mapping extends the target space of kinase inhibitors. *bioRxiv*. doi: 10.1101/2019.12.31.891812

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems 28* (pp. 2224–2232). Curran Associates, Inc. Retrieved from `http://papers.nips.cc/paper/5954-convolutional-networks-on-graphs-for-learning-molecular-fingerprints.pdf`

Feng, Q., Dueva, E., Cherkasov, A., & Ester, M. (2018). *Padme: A deep learning-based framework for drug-target interaction prediction.*

He, T., Heidemeyer, M., Ban, F., Cherkasov, A., & Ester, M. (2017, April). Simboost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines. *Journal of Cheminformatics*, *9*(1), 24. doi: 10.1186/s13321-017-0209-z

Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Félix, E., ...

Leach, A. R. (2018, 11). ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, *47*(D1), D930-D940. doi: 10.1093/nar/gky1075

Nguyen, T., Le, H., & Venkatesh, S. (2019). Graphdta: prediction of drug–target binding affinity using graph convolutional networks. *bioRxiv*. doi: 10.1101/684662

Open-source cheminformatics [Computer software manual]. (n.d.). Retrieved from `http://www.rdkit.org`

Pahikkala, T., Airola, A., Pietilä, S., Shakyawar, S., Szwajda, A., Tang, J., & Aittokallio, T. (2014, 04). Toward more realistic drug–target interaction predictions. *Briefings in Bioinformatics*, *16*(2), 325-337. doi: 10.1093/bib/bbu010

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Édouard Duchesnay (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(85), 2825-2830. Retrieved from `http://jmlr.org/papers/v12/pedregosa11a.html`

Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., & Pande, V. (2015). *Massively multitask networks for drug discovery.*

Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742-754. (PMID: 20426451) doi: 10.1021/ci100050t

Sliwoski, G., Kothiwale, S., Meiler, J., & Edward W. Lowe, J. (2014). Computational methods in drug discovery. *Pharmacological Reviews*, 336-386. doi: 10.1124/pr.112.007336

Wallach, I., & Heifets, A. (2018). Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of Chemical Information and Modeling*, *58*(5), 916-932. (PMID: 29698607) doi:

10.1021/acs.jcim.7b00403

Öztürk, H., Özgür, A., & Ozkirimli, E. (2018, Sep). Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, *34*(17), i821–i829. doi: 10 .1093/bioinformatics/bty593