IDENTIFYING NUCLEAR RECEPTOR LIGANDS THROUGH SEQUENCE-BASED DEEP LEARNING

A Thesis

by

FANGTONG ZHOU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Yang Shen
Committee Members,	Xia Hu
	Krishna Narayanan
	Xiaoning Qian
	Chao Tian
Head of Department,	Miroslav M. Begovic

May 2020

Major Subject: Electrical Engineering

ABSTRACT

This project focuses on developing machine learning methods to predict protein-ligand interactions. The unique proteins under study are nuclear receptors (NRs), which regulate hormone-triggered gene transcription and are often drug targets in cancer therapy. Compared to other proteins, the categorical labels for their ligand interactions are much more complex to annotate and learn in the framework of machine learning. The project aims at identifying ligands for NRs through sequence-based deep learning while addressing aforementioned challenges.

The main contributions of this project include the following. (1) Data curation: Identification and curation of databases were performed. A rule was set up to deal with the complicated categorical labels for NR-ligand pairs. (2) Machine Learning Models: Shallow models, two-step deep model, and jointly trained deep model were trained. (3) Stratified Validation Sets: They were developed to tune the hyper-parameter of the model and improve model generalizability. (4) Transfer Learning: It was applied to tune models trained on other NRs so that novel ligands can be identified for orphan NRs.

Specifically, categorical labels were first collected and curated for the identified data sets to enable model training and testing. Protein and ligand features were extracted by a pre-trained recurrent neural network (RNN) encoder using unlabeled data and then fed to various downstream supervised models, shallow or deep, for multi-class classification. Among shallow supervised models random forest showed the best results. For deep supervised models, a convolutional neural network (CNN) was trained subsequently or jointly with RNN. Comparisons between various shallow and deep models showed that although the way to train deep models, separately or jointly, did not make significant difference in model performance, there was an obvious improvement from shallow to deep models. Moreover, a stratified validation strategy was developed to further improve the generalizability of the model from the training set to test sets. Lastly, considering the very different distributions of biological features between training NRs and orphan NRs, transfer learning strategy was used to fine tune the model and improve the performance of ligand identifi-

cation for two orphan NRs. Future plan includes the exploration of mutational effect, that is, the change in predicted label upon amino-acid substitutions, insertions, or deletions in NRs.

DEDICATION

I dedicate this dissertation work to my advisor Dr. Shen who have guided me, taught me and offered great help for me during the whole process.

I also dedicate this dissertation work to my loving mother who is far away across the Pacific Ocean and yet still encouraged and comforted me when I was depressed.

I dedicate this work and give special thanks to my lab group members especially Di and Mostafa. Without them I can never accomplished this work by myself.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Yang Shen, Professor Krishna Narayanan, Professor Xiaoning Qian and Professor Chao Tian of the Department of Electrical and Computer Engineering and Professor Xia Hu of the Department of Computer Science and Engineering.

The data analyzed for this project was provided by two databases: Nuclear Receptors Database Including Negative Data and Orphan Nuclear Receptor Ligand Binding Database.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This work was completed without outside financial support.

NOMENCLATURE

NRs	Nuclear Receptors
NRLiSt BDB	NRs ligands and structures benchmarking database
NR-DBIND	Nuclear Receptors Database Including Negative Data
ONRLDB	Orphan Nuclear Receptor Ligand Binding
SPS	Structural property sequence
SMILE	Simplified molecular-input line-entry system
SVM	Support Vector Machine
RF	Random Forest
KNN	K-Nearest Neighbors
RNN	Recurrent neural network
CNN	Convolutional neural network

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Background knowledge for Nuclear Receptors

This project intends to solve machine learning problems of a particular kind of protein, nuclear receptors, which are transcription factors that regulate gene expression in various physiological processes through their interactions with small hydrophobic molecules. For this project, two tasks have been managed to finish.

First, machine learning models were developed to predict protein-ligand binding. The optimal model among all of the model used was chosen. The data used was collected from the NR-DBIND database which will be discussed later. I used NRs with their binding ligands to train machine learning models and then test them to chose the one with the best performance. Next, transfer learning was used to find ligands that bind with a particular kind of NRs, orphan receptors, which has few ligands bind with them. I used the model with the best performance in last step to test adopted orphan receptors not seen in the last step and apply transfer learning on this problem to improve. The first step optimal model was used as a pre-trained model. A new training data set was used to fine tune the model. The data used was collected from the ONRLDB database which will be introduced later. Last, how the mutation in protein will affect the interactions between protein and ligands will be explored and tested using the previous model for a future plan study.

In this project, plenty of the computational skills are performed and methods used are based on biology consideration. Machine learning method, in a more specific way, identifying NRs ligands through sequence-based deep learning is developed and applied to this biology problem and so that this problem can be illustrated in a more effective and rational computational angle. Therefore, it is quite clear that the significance of applying computational methods to biology problem in both field of study.

1.1.1 Nuclear Receptors

Nuclear Receptors(NRs) distinct in their structural organization and are targets for drugs and endocrine disruptors; thus, it is important to understand and predict how molecules selectively bind NRs and how their binding triggers a specific activity. [2]

There are two kinds of NRs: endocrine receptors, orphan receptprs. Endocrine receptors possess hormones as their endogenous ligands; orphan receptors have no known or universally agreed upon endogenous ligands. NRs are potential drug targets because of their functions in human body. Due to the fact that endocrine NRs have large ranges of ligands that binds with them, the use of the endocrine NRs will cause severe side effects. Belonging to the same super family of NRs, orphan NRs can be potential drug targets since they have none ligands found binding to them.

1.1.2 Three databases

There are three databases that this project consulted.

(1) NRLiSt BDB: NRs ligands and structures benchmarking database[2]

This database includes 27 NRs (out of 48 NRs in total), 339 protein structures, 9905 ligands. It uses structure-based and ligand-based virtual screening methods.

(2) NR-DBIND: Nuclear Receptors Database Including Negative Data[3]

This database includes 28 NRs (out of 48 NRs in total), 593 protein structures, 7593 ligands, 15116 affinity data for 13566 unique interactions. It uses holo structure-based, IEC50-based, maximum activity-based methods.

(3) ONRLDB: Orphan Nuclear Receptor Ligand Binding Database[4]

This database includes 34 NRs (out of 48 NRs in total), 11000 ligands. It uses structure search, advanced search, clustering tool, tree view.

1.1.2.1 Choice of the Database

In this project, all the data used is provided by NR-DBIND and ONRLDB databases for the following reasons. For the first step, this project intends to focus on the endocrine NRs and adopted orphan NRs in order to have more protein-ligand pairs available. Therefore more data could be

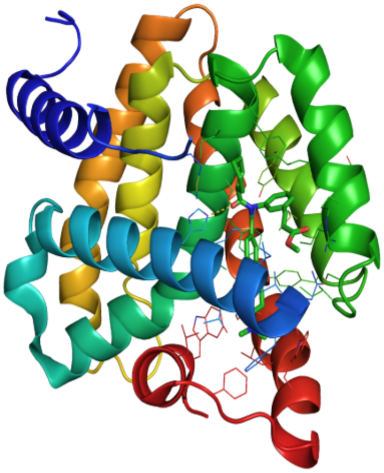

Figure 1.1: A cartoon representation of a 3D protein structure of Farnesoid X-activated receptor (FXR). A kind of adopted orphan NRs, also known as Nuclear receptor subfamily 1 group H member 4.

included in the training set and thus the model can capture more of the distribution of the features of the nuclear receptors. Then the optimal model can be used for other purpose. Considering the possibly different distributions of biological features between orphan NRs and the entire population of all NRs, transfer learning strategy was used for next step to test binding ligands for the orphan NRs.

For the second step of the project, ONRLDB will be used to have a further step about the predictions for the orphan NRs with few ligands found binding with them. Orphan NRs which are not used in the previous training model will be used to qualify the performance of the model. These orphan NRs will have the ground truth binding ligands, however, with few numbers and this situation would be appropriate to see whether the transfer learning can help the model to capture the new features or not.

1.1.2.2 Content of NR-DBIND

There are in total 28 NRs in this database. In the first place, depending on the pharmacological profile of the binding ligand of one of the NRs, different types of activities are illustrated. There are some definitions need to be clarified.

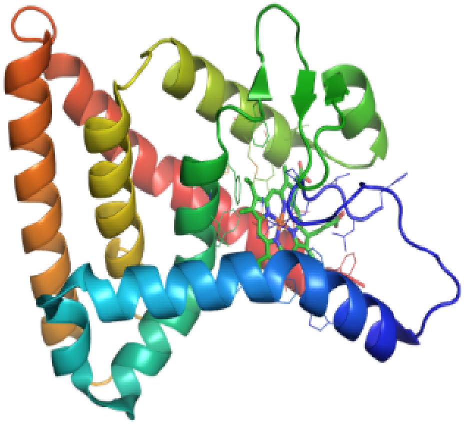

Figure 1.2: A cartoon representation of a 3D protein structure of Rev-erb alpha related receptor (Rev_erb_beta). A kind of orphan NRs, also known as Nuclear receptor subfamily 1 group D member 2.

(1) Agonist compound: they inhibit NRs basal unbound transcription activity.

(2) Partial agonist compounds: associated with an incomplete physiological response.

(3) Inverse agonist compounds: trigger the opposite physiological response compared to agonist compounds.

(4) Antagonist compounds: block NRs in their inactive conformation by impeding the recruitment of co-activators and favoring interactions with co-repressors.

(5) Partial antagonist compounds: associated with an incomplete physiological response.[2]

An illustration demonstrating the response of different kinds of drugs is shown as below. More details will be presented in the next chapter mainly describing about data.

1.2 Contributions

The main contributions of this project are:

I made an identification of databases used and curation of data sets including protein and ligands. The formation of the inputs are considered and eventually I made both of the inputs of the following models linear forms. For the protein sequence, a method called SPS[5] from Deep Affinity was used to curate the ligand binding domain of the sequence. For the compound format,

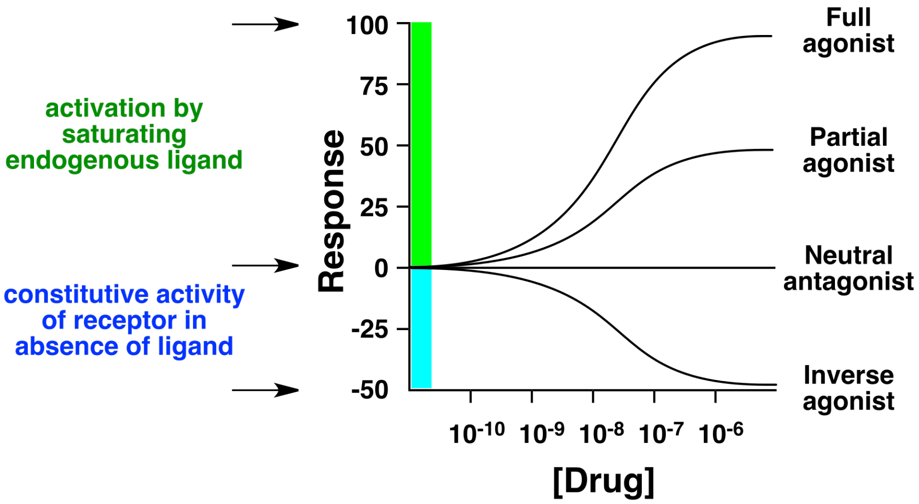

Figure 1.3: An illustration that demonstrates the response of different kinds of drugs. Reprinted from[1]

canonical SMILE format was used as the simplified molecular input format for the ligands used. Shallow models, two-step deep model, jointly trained deep model, stratified jointly trained deep model were applied then.

To solve the first challenge about the input data curation, a unique rule to decide the labels for the binding was developed. Due to the five kinds of binding provided by the data bases, I decided to treat this problem as a classification problem. The categorical labels are much more complex than the other kinds of protein studied. To decide the annotation to use from the database and to develop the rules for label of each protein-ligand pair are one of the contributions made in this project.

To make an improvement when tuning the hyper-parameter of the model, a stratified way of splitting the origin data set was developed. In this project for the first step, I will need to generalize the models from the training set to the test sets. Because of the different test sets being set up, a similar way of splitting the validation set was used. Four different kinds of the data have been split here in this project: protein unique, ligand unique, both unique, both not unique. Data have been split into different test sets to show the adaptability of the model. To make the model more general, the strategy validation was developed to solve this problem. This would be another contribution in this project.

After training the model, transfer learning would be applied to see how the model will generalize for the orphan NRs. Using the adopted orphan NRs with few ligands binding to fine tune the optimal model from the last step. This would be the basis for the possible exploration of finding new ligands for these adopted orphan NRs, or even for those orphan NRs that are found binding with no ligands. This would be significant for finding new drugs to target at those orphan NRs. This would be the last contribution of this project.

2. DATA CURATION

2.1 Data Overview

For the first step, based on the NR-DBIND database, two kinds of Ligand Pharmacological Profile Annotation are provided. The name of the protein and the SMILE of the compound are also provided. That is to say NR-DBIND present two ways to classify the protein-compound binding affinity.

2.1.1 IEC50 classification

In NR-DBIND, this classification is based on IC50 and/or EC50 activity data. A compound is recognized as:

(1) 'agonist' if a finite EC50 has been experimentally measured in an agonist activity evaluation assay;

(2) 'antagonist' if it is a finite IC50 in an antagonistic activity evaluation assay;

(3) 'agonist/antagonist' if both have been measured.[2]

2.1.2 The maximum activity set

In NR-DBIND, this classification accounts for the experimentally measured percentage activity as compared to a reference compound (p.a.r.) when it reaches a plateau on the dose-response curve. A compound is recognized as:

(1) In agonist mode,

'agonists' if ligands displaying >= 75 p.a.r.

'partial agonists' if ligands displaying between >= 25 and < 75 p.a.r.

'inverse agonists' if ligands displaying < -25 p.a.r.

(2) In antagonist mode,

'antagonists' if ligands displaying >= 75 p.a.r.

'partial antagonists' if ligands displaying between >= 25 and <= 75 p.a.r.

Ligands displaying between -25 and 25 p.a.r. were not annotated because of their weak activity.[2]

2.1.3 Choice of the classification method

Here in this project, I chose the maximum activity set annotation, for the following reasons:

(1) The the maximum activity set annotation has a more comprehensive description for the ligands attribution. It use partial to capture the degree of the binding between protein and ligands. EC50 or IC50 may indicate that an agonist or an antagonist activity is detected, but it may not be sufficient to determine the pharmacological profile of a ligand. It also concludes Inverse agonist which are not found in the other annotation.

(2) For the data available, the different modes and reference compounds are not comparable and may be even different for the fact that the data are collected from different papers and sources. I decided to make this problem a classification one so that the labels could be used comparable. The experimentally measured percentage data will be not used here in this project.

2.1.4 Function of ligand

For the ONRLDB, the function of the compound is provided.The name of the protein and the SDF form of the compound are also provided. When protein not shown in the NR-DBIND, there are two kinds of protein found available in the data set and the function of the ligands available is either ’Agonist’ or ’Antagonist’. This would agree with what has been discussed above.

2.2 Curation of the data

For the classification problem here, proper forms of the proteins(NRs) and ligands need to be decided. These are the input feature needed for the problem. The labels of the protein-ligands pairs are needed as the input Label for the machine learning models. These labels will be derived from the annotation mentioned above, the maximum activity set annotation.

After downloading the raw data, I extracted the protein names, ligand SMILES and kinds of the binding to use them as the raw input of my project. And there are some pre-processing steps of the data that need to be done.

(1) Protein input formation: use the method learned in the Deep Affinity project. Check every protein in the Uniprot database, find the ligand binding domain of the sequence of this unique kind

of NR and use the method in Deep Affinity to transfer those ligand binding domain sequence into Structural property sequence (SPS) representation form.[5] The form was developed because of the expense and lack of 3D protein structure.[6] Also, the prediction of the 3D structure could be of low precision without a template structure. Therefore, in this project, I also choose this method to identify the ligands binding via sequence-based model. The Scratch Protein Predictor was used to get the SSPro(Secondary Structure) and ACCpro(Solvent Accessibility) file of the particular kind of the protein.[7] These two formations were used to generate the SPS form with high resolution and interpret ability.

(2) Ligand input formation: make sure every SMILE format used is the canonical SMILE (simplified molecular-input line-entry system) [8] in the Pubchem database, which is a more simple line notation using short ASCII strings. It described the structure of chemical species. There are cases that one compound and its chiral structure are of the same canonical SMILE. For such cases, I choose to use the same format for these two compounds.

(3) Labels: for the NR-DBIND, because there are several ways of deciding the labels, there is a protocol I have decided to make sure of the label for every pairs as following after consulting with the original database collectors.

(a) transferred perc annotation: this annotation transferred all the different activity annotations from different papers and it give the simple level of 'Agonist' or 'Antagonist'. There can be conflicts between this annotation and perc based annotation;

(b) perc based annotation as the main criterion:this annotation percentage activity gives more precision on the level of activity reached in the paper referred. An 'Agonist' in the transferred perc annotation can be an 'Agonist' or an 'Partial Agonist'. An 'Antagonist' in the transferred perc annotation can be an 'Antagonist' or an 'Partial Antagonist'. There can be conflicts between this annotation and transferred perc annotation;

(c) label conflicts between (a) and (b): (i) transferred perc annotation gives multi label for one pair while perc based annotation gives one: choose perc based annotation as the label; (ii) transferred perc annotation and perc based annotation do not agree with each other: choose the two-thirds majority label for this pair or else delete this pair;

(d) replicate pairs with different labels: this is because the data are manually collected from different sources, thus there can be results that do not agree with each other. I made the following decisions: (i) choose the two-thirds majority label for this pair; (ii) If there is a tie, choose the label with smaller number of data sample in the class distribution for the whole data set.

For the ONRLDB, the function of the compound is provided as either ’Agonist’ or ’Antagonist’. This would agree with the rules have been discussed and set up above.

Two flow charts are provided to have a clear view of this rule, take tran as the abbreviation of the transferred perc annotation and perc as the abbreviation of the perc based annotation.

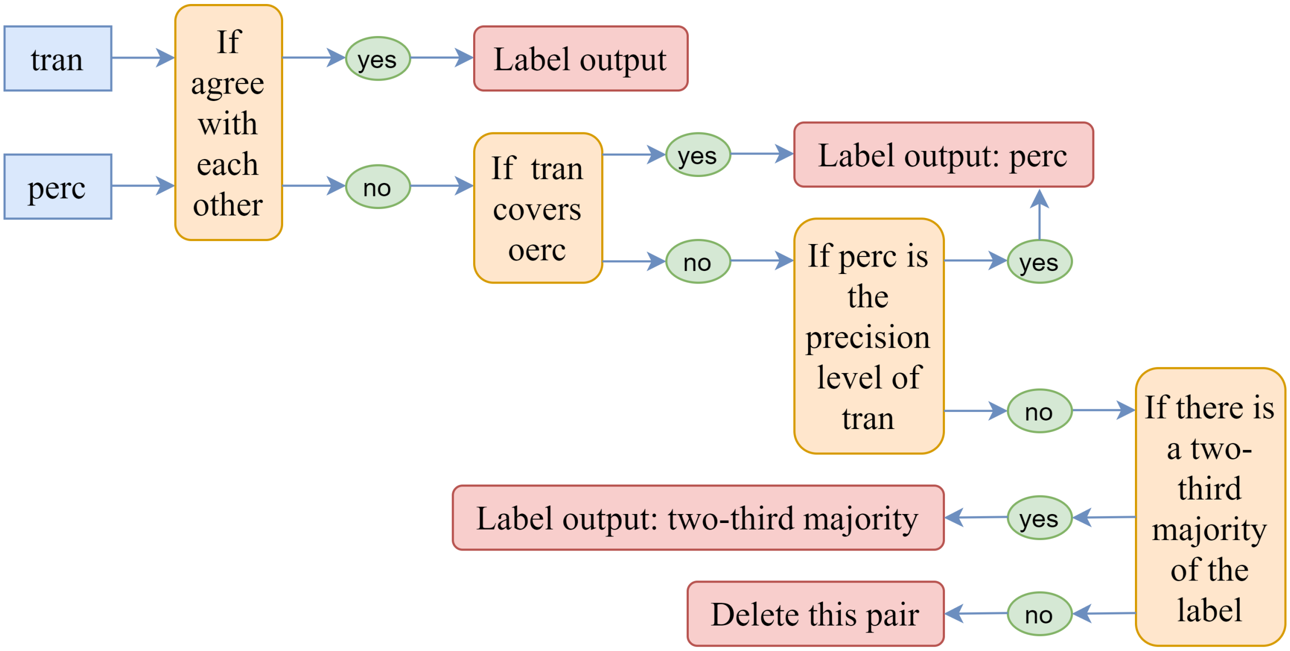

Figure 2.1: Principle 1 to decide the labels

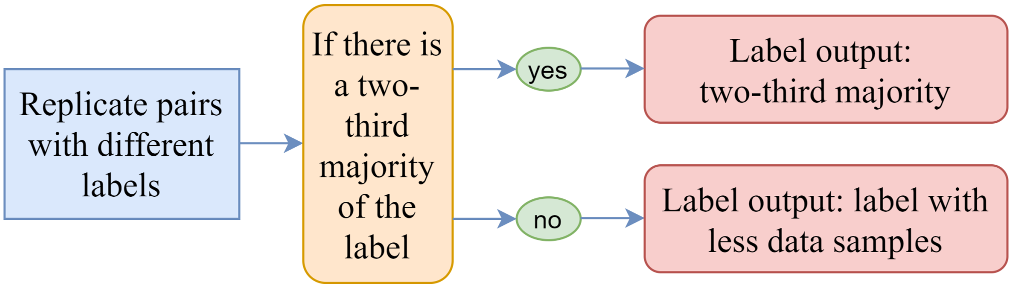
Replicate pairs with different labels
If there is a two-third majority of the label
yes
no
Label output: two-third majority
Label output: label with less data samples

Figure 2.2: Principle 2 to decide the labels

2.3 Description of the Data

2.3.1 NR-DBIND

How the proteins or ligands will affect the model predictions ability and how sensitive is the model for the protein SPS formation and the canonical SMILE formation are of much importance.Therefore, I split the test sets into the following four kinds:

(1) Test0: training set has the kinds of proteins and the kinds of ligands in this test set (protein and ligand are not unique)

(2) Test1: training set has the kinds of proteins but not the kinds of ligands in this test set (ligands are unique)

(3) Test2: training set does not have the kinds of proteins but has the kinds of ligands in this test set. (proteins are unique)

(4) Test3: training set does not have the kinds of proteins or the kinds of ligands in this test set (proteins and ligands are both unique)

Here is a table that describes the data distribution of the data collected from NR-DBIND after being pre-processed by the steps illustrated in the Curation of the data, labels part:

Here we can see the lack of data points in the ’Inverse agonist’ and ’Partial antagonist’. Therefore, the results analysis will mainly place the emphasis on the other three class: ’Agonist’, ’Partial agonist’, ’Antagonist’.

	Agonist	Partial agonist	Inverse agonist	Antagonist	Partial antagonist
Total	1065	1108	80	379	175
Train	698	680	71	275	136
Test0	12	13	-	7	-
Test1	293	324	5	67	24
Test2	48	56	3	24	11
Test3	14	35	1	6	4

Table 2.1: data distribution of NR-DBIND

2.3.2 ONRLDB

In this database, there are only two kinds of protein that are not shown in the NR-DBIND and have ligands binding with them as well.

(1)PXR: Nuclear receptor subfamily 1 group I member 2

(2)Rev_erb_alpha: Nuclear receptor subfamily 1 group D member 1

Here is a table that describes the data distribution of the two kinds. These two kinds of orphan NRs will be used together as an orphan test set to test the generalization of the model with and without transfer learning.

	Agonist	Antagonist
PXR	8	21
Rev_erb_alpha	23	0

Table 2.2: data distribution of ONRLDB

3. METHODS

3.1 Criterion and evaluation methods

3.1.1 The area under the precision-recall curve (AUPRC)

Because of the imbalanced data in the test set and the emphasis should be paid on the cases that are predicted correctly, AUPRC was the first metric considered to be used. Usually this methods will be used for a binary classification case where threshold could just simply be 0.5. However, due to the multiple classes in the data set, the threshold for the probability for each of the five classes can hardly be the same (not simply 0.2). When choosing the highest probability of the five prediction outcomes for each data point, it can be the situation that the largest probability outcome for data point A is smaller than the second largest probability outcome for data point B. Therefore, this will not be a reasonable metric to use.

Another adjustment for the method to use AUPRC to fit the five classes situation has also been considered. This task can be considered as many different binary classification problems. AUPRC need to be calculated as class0 vs. not class 0, class1 vs. class1 and etc. However, this method will have the same problem as the previous one discussed above. The same threshold cannot be found. It is unreasonable to convert this problem into many binary problems. Therefore, I decided to apply other reasonable metrics to evaluate the results.

3.1.2 F1 score

Since the above reasons stated and the consideration of the emphasis on the predicted correctly cases, the F1 score is decided to use. Since F1 score considers both recall and precision as follow:

$F_1 = (\frac{2}{recall^{-1}+precision^{-1}}) = 2 * \frac{precision*recall}{precision+recall}$

Therefore, the results of each of the classes can be compared and evaluated in a more comparable way. This metric is then used to tune the hyper-parameters of the models and to evaluate the performance of the models used.

For shallow model, the optimal hyper-parameters would be the one with the largest average

F1 score of the five classes after the cross validation process. It has also been made sure that the metric for the cross validation is also the largest average F1 score of the five classes so that the tuning part would make more sense. Then every shallow model can be trained using the optimal hyper-parameters. The optimal model would be the one with the largest average F1 score of the five classes of all the test sets.

For deep model, the optimal hyper-parameters would be the one with the largest average F1 score of the five classes of the validations set or the strategy validation sets. The tuning part would be a little bit more complex than the shallow model. The hyper-parameters need to be tuned one at each time and when tuning a particular one, all the others need to be fixed to decide this optimal hyper-parameter. Then all of the hyper-parameters can be tuned by doing so. The optimal model would be decided using a paired t-test as discussed below. It cannot be simply considered the optimal model for every test set by checking the the largest average F1 score of the five classes of the test sets, since there can be much difference among the four test sets. Therefore, four test sets will be treated separately and a paired t-test will be used for each of the test sets and the four test sets will be applied a paired t-test as well to decide the best model for each of the test sets separately or in a whole performance of the four.

Based on this particular project, the baseline for the F1 score shall be the random case F1 score: $F_1 = (\frac{2}{recall^{-1}+precision^{-1}}) = 2 * \frac{precision*recall}{precision+recall} = (2 * \frac{0.2*0.2}{0.2+0.2}) = 0.2$

3.2 Shallow Models

For this classification problem, the task is to predict the protein-ligand interactions. The inputs would be SPS form of proteins, canonical SMILE of ligands and labels for each pair. Support Vector Machine (SVM), Random Forest(RF), k-nearest neighbors (KNN) are chosen for this classification problem as the downstream models.

(1) Inputs: Use the SPS and canonical SMILE as the inputs and feed them to a pre-trained RNN encoder which has been through an unsupervised training by deep learning with abundant unlabeled data. And then the extracted feature matrix can be obtained. This matrix will be the input for the shallow models.

(2) Data pre-processing: the training data is highly imbalanced. To balance every class, use oversampling or SMOTE to make the number of each class the same.

The following is the data distribution of the five classes:

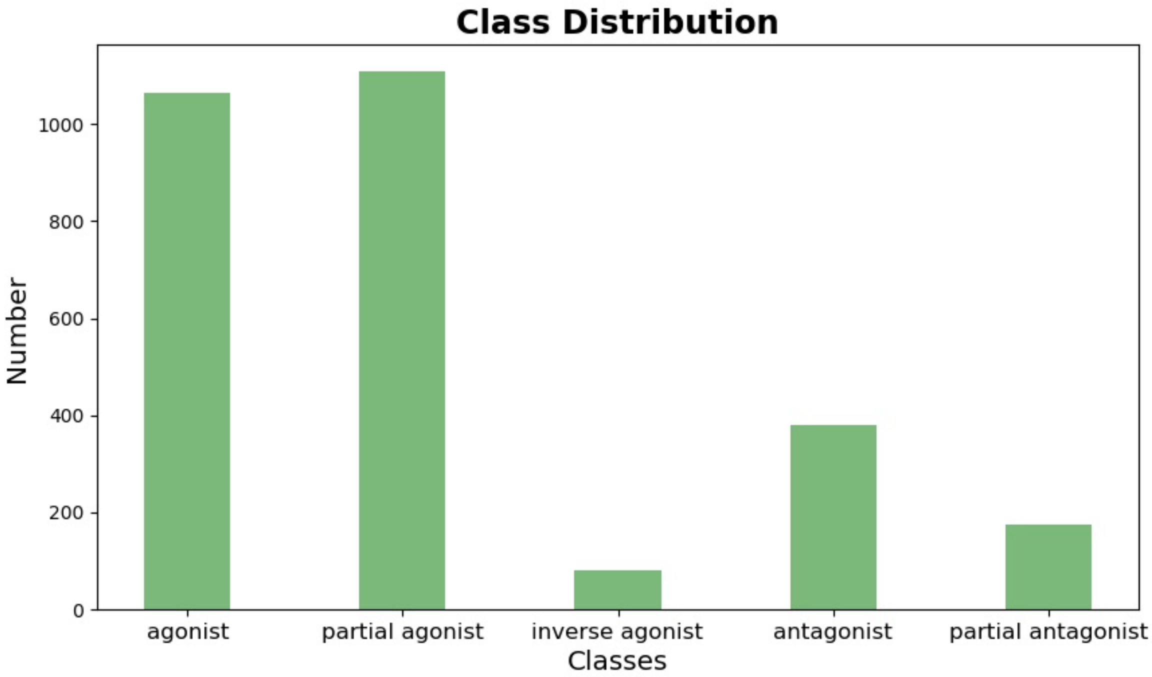

Figure 3.1: The data distribution of the five classes

(3) Hyper-parameter tuning: use cross validation to tune the hyper-parameter and the criteria is decided to be the largest average F1 score of the five classes. After choosing the optimal hyper-parameters of one particular model. All the test data sets are fed to the shallow models and F1 score is used again to decide the shallow model with the best performance.

The following is the structure of shallow model:

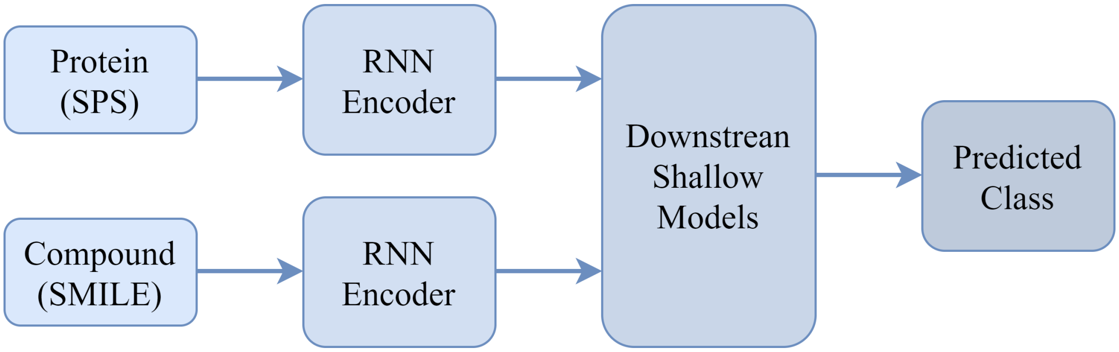

Figure 3.2: The pipeline of shallow model method to predict the protein-ligand binding

3.3 Deep Models

3.3.1 Two-step Deep Model

In order to have a better performance, a deep learning model is used. A unified recurrent neural network-convolutional neural network (RNN-CNN) [5] was used and was trained in a two-step way, which means to first train the RNN encoder and then fixed the parameter of the encoder. Here, the RNN model used is sequence-to-sequence (seq2seq). Sequence-to-sequence (seq2seq) models[9] have enjoyed great success in a variety of tasks such as machine translation, speech recognition, and text summarizing.[10] Here in this project, it is used to embed the SPS and SMILE formation into feature matrix. This feature matrix was then the input of the CNN model.[11] Then, train the CNN follows. A two-step way is made to be comparable with the shallow model which is also trained in two steps.

(1) Inputs: SPS form of the proteins, canonical SMILE of the ligands.

(2) Data pre-processing: the training data is highly imbalanced. To balance every class, use oversampling to make the number of each class the same.

(3) Hyper-parameter tuning: use a randomly split validation set to tune the hyper-parameter and the criteria is decided to be the largest average F1 score of the five classes of validation set.

The following is the structure of deep model:

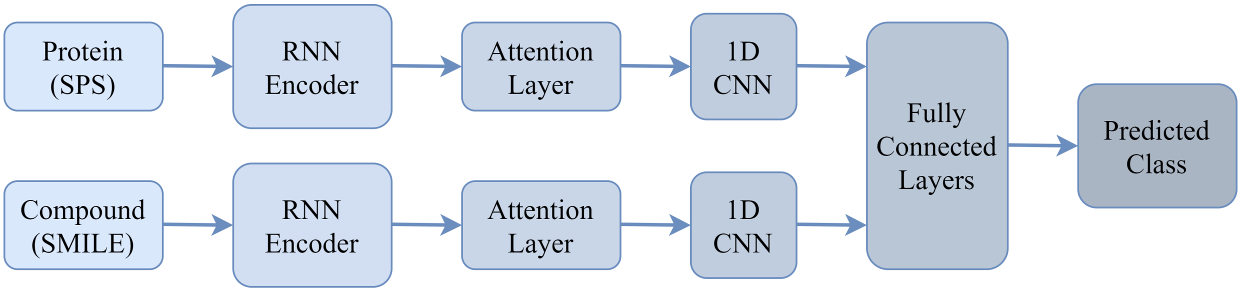

Figure 3.3: The pipeline of deep model method to predict the protein-ligand binding

3.3.2 Jointly-trained Deep Model

In order to have a better performance, I used the unified RNN-CNN and trained this model in a jointly way, which means to train the RNN and the CNN follows together without fixing the parameter of the RNN encoder. A jointly-trained way is made to be comparable with the two-step way deep model to decide whether two-step way or one-step way training method is better.

(1) Inputs: SPS form of the proteins, canonical SMILE of the ligands.

(2) Data pre-processing: the training data is highly imbalanced. To balance every class, use oversampling to make the number of each class the same.

(3) Hyper-parameter tuning: use a randomly split validation set to tune the hyper-parameter and the criteria is decided to be the largest average F1 score of the five classes of validation set.

3.3.3 Stratified Jointly-trained Deep Model

In order to shorten the distance between the training set and the test sets, I used the unified RNN-CNN and trained this model in a jointly-trained way with four validation sets which are split manually. The four validation sets are split like the test sets:

Validation0: training set has the kind of protein and ligand in this test set

Validation1: training set has the kinds of proteins but does not have the kinds of ligands in this validation set

Validation2: training set does not have the kinds of proteins but has the kinds of ligands in this validation set

Validation3: training set does not have the kinds of proteins or the kinds of ligands in this validation set

(1) Inputs: SPS form of the proteins, canonical SMILE of the ligands.

(2) Data preprocessing: the training data is highly imbalanced. To balance every class, use oversampling to make the number of each class the same.

(3) Hyper-parameter tuning: use a strategy split validation sets to tune the hyper-parameter and the criteria is decided to be the largest average F1 score of the five classes of all the validation sets.

3.4 Transfer Learning

The second step of this project is to use transfer learning[12] to generalize the optimal trained model to another kind of data set, orphan NRs. 50percent of the data set is set to be the test set of the model. First, the original trained optimal model is used directly to see how the model trained with the NRs will generalize from one data set to another. Next, use the other half of the orphan data set to fine-tune the pre-trained model. The first embedding layer and the last half of the fully connected layers were fine-tuned. The RNN encoder, attention layer and the CNN layer of the model were fixed. Also, different percentage of the orphan data set was used. 50%, 40%, 30%, 20%, 10% and 6% of the orphan data set were applied to the fine-tune part and it has been made sure that the larger training data set includes the smaller one.

(1) Inputs: SPS form of the proteins, canonical SMILE of the ligands.

(2) Data preprocessing: the training data is highly imbalanced. To balance every class, use oversampling to make the number of each class the same.

(3) Test set performance: the F1 score of of the different strategy were compared.

3.5 Mutation for Future Exploration

Chose the model with the best performance among all the above shallow and deep models, train this model with all the data combined, use the mutation data of as the test set. This part is designed to find out if a mutation in the protein sequence will lead to the binding ligand change from one class to another. [13][14]

4. RESULTS AND CONCLUSIONS

4.1 Results of the shallow and deep models

After the hyper-parameter tuning part, the optimal combination of the hyper-parameters for the shallow model and the two-step deep model, jointly-trained deep model and jointly-trained deep model with strategy are set. These models were then trained with these optimal combination of the hyper-parameters and the previous training, validation and test sets all together to generalize the models to the largest degree.

After training the models, the test sets were used to make the predictions and see how the models actually work for the data. Paired t test shall be made to quantify the performance of the model for different test sets and will be the basis for the next steps of the orphan receptors and mutation part.

4.1.1 Results of the F1 score tables

Here are the results of the shallow models and deep models :

(1) the optimal shallow model is Random Forest with the method of SMOTE:

F1 score	Agonist	Partial agonist	Inverse agonist	Antagonist	Partial antagonist
Train	0.6861	0.7025	0.9701	0.8602	0.8848
Test0	0.6090	0.5833	-	0.7692	-
Test1	0.0891	0.6103	0.0	0.0417	0.0
Test2	0.6731	0.3659	0.4444	0.5	0.0
Test3	0.3077	0.6250	0.0	0.0	0.0

Table 4.1: Random Forest

(2) the result of the two-step deep model:

F1 score	Agonist	Partial agonist	Inverse agonist	Antagonist	Partial antagonist
Train	0.9231	0.9305	0.9979	0.9745	0.9787
Test0	0.6667	0.7083	-	0.6667	-
Test1	0.4718	0.4486	0.2000	0.2483	0.1333
Test2	0.59578	0.6286	0.5000	0.6000	0.2400
Test3	0.2222	0.4590	0.0	0.2727	1.0

Table 4.2: Two-Step RNN-CNN

(3) the result of the jointly-trained deep model:

F1 score	Agonist	Partial agonist	Inverse agonist	Antagonist	Partial antagonist
Train	0.9405	0.9582	0.9986	0.9792	0.9796
Test0	0.4348	0.5714	-	0.8333	-
Test1	0.3902	0.5168	0.1724	0.3091	0.0
Test2	0.6400	0.5606	0.8000	0.6053	0.2857
Test3	0.1463	0.4789	0.0	0.4000	0.0

Table 4.3: Jointly Trained RNN-CNN

(4) the result of the stratified jointly-trained way deep model:

F1 score	Agonist	Partial agonist	Inverse agonist	Antagonist	Partial antagonist
Train	0.9314	0.9484	0.9986	0.9769	0.9770
Test0	0.5600	0.4800	-	0.6667	-
Test1	0.3876	0.5062	0.5000	0.3506	0.2083
Test2	0.6906	0.6714	0.8000	0.5676	0.2308
Test3	0.1579	0.5294	0.0	0.4000	0.0

Table 4.4: Stratified Jointly Trained RNN-CNN

4.1.2 Results of the F1 score bar plots

To make the results more visible and direct, here are five bar plots describing the F1 score of the data sets for the four test sets among four kinds of models: random Forest, two-step deep model, joint-trained deep model and joint-trained deep model with strategy validation sets. Each of the bars represents an F1 score for one test sets. The random F1 score is 0.2. The criterion for the test sets is the average value of the F1 scores of the five classes.

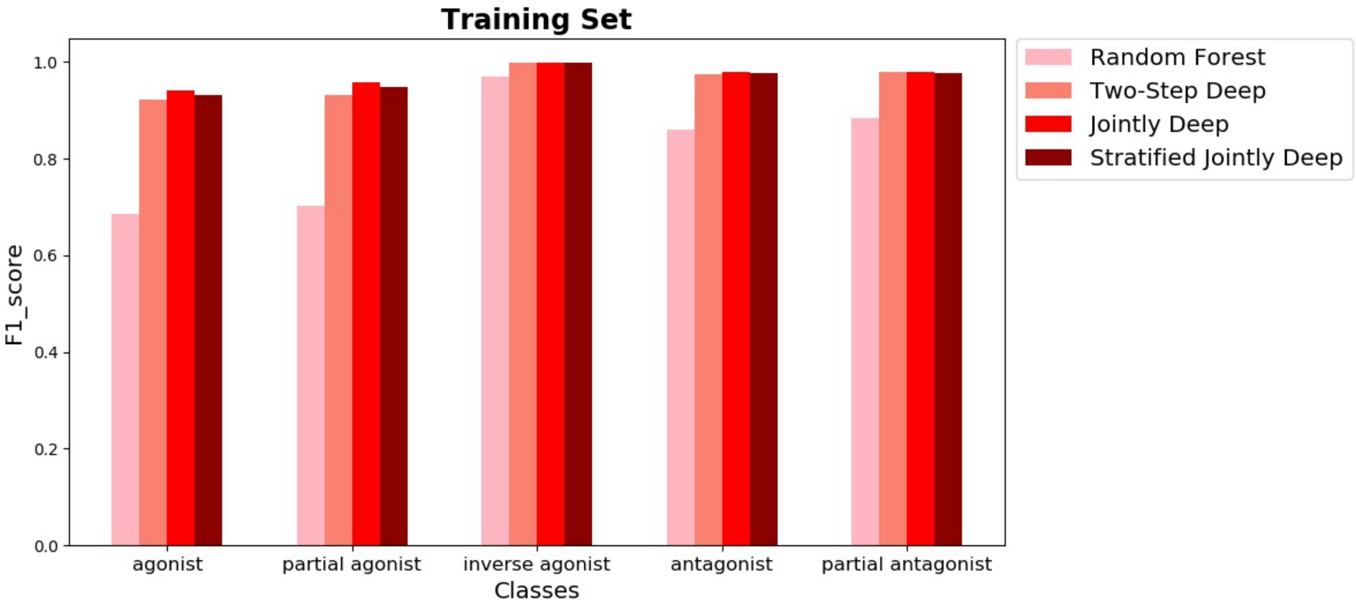

Figure 4.1: F1 score of the training set for the four test sets among four kinds of models

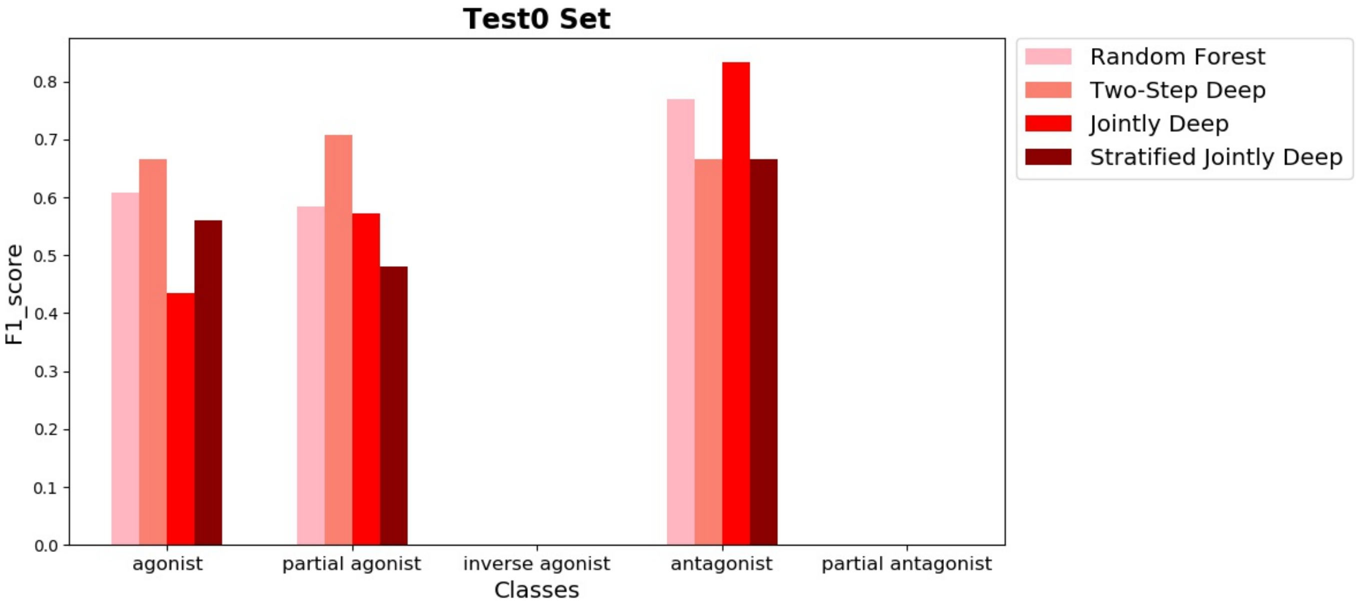

Figure 4.2: F1 score of the Test0 set for the four test sets among four kinds of models

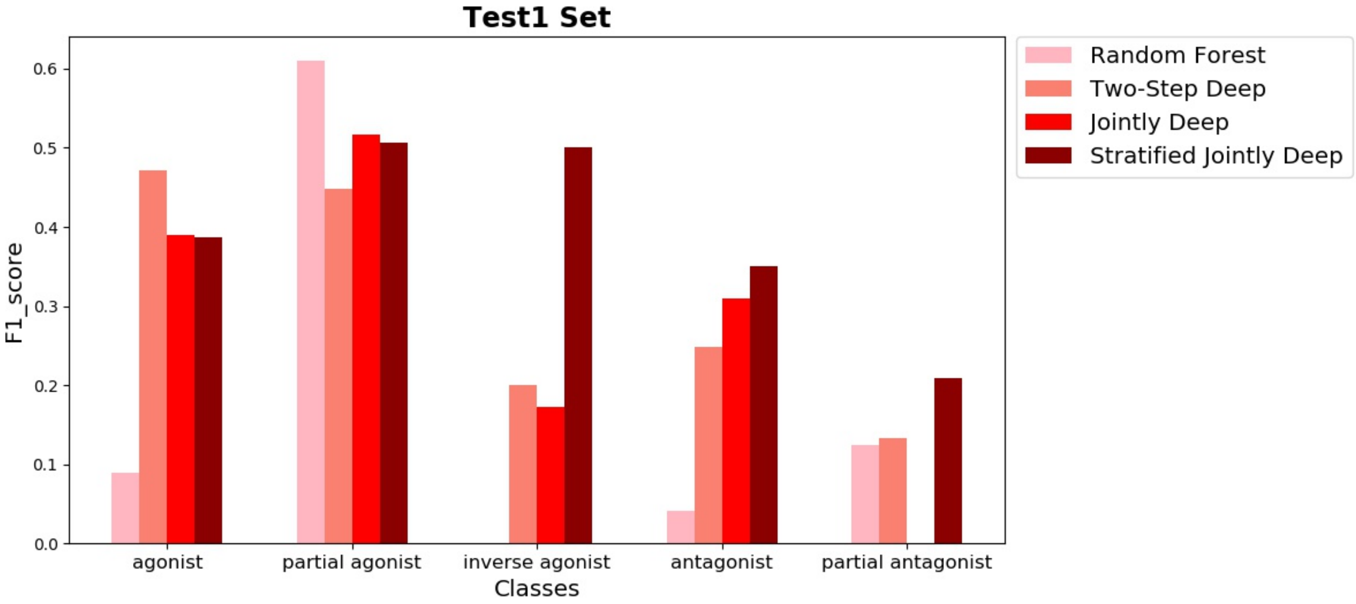

Figure 4.3: F1 score of the Test1 set for the four test sets among four kinds of models

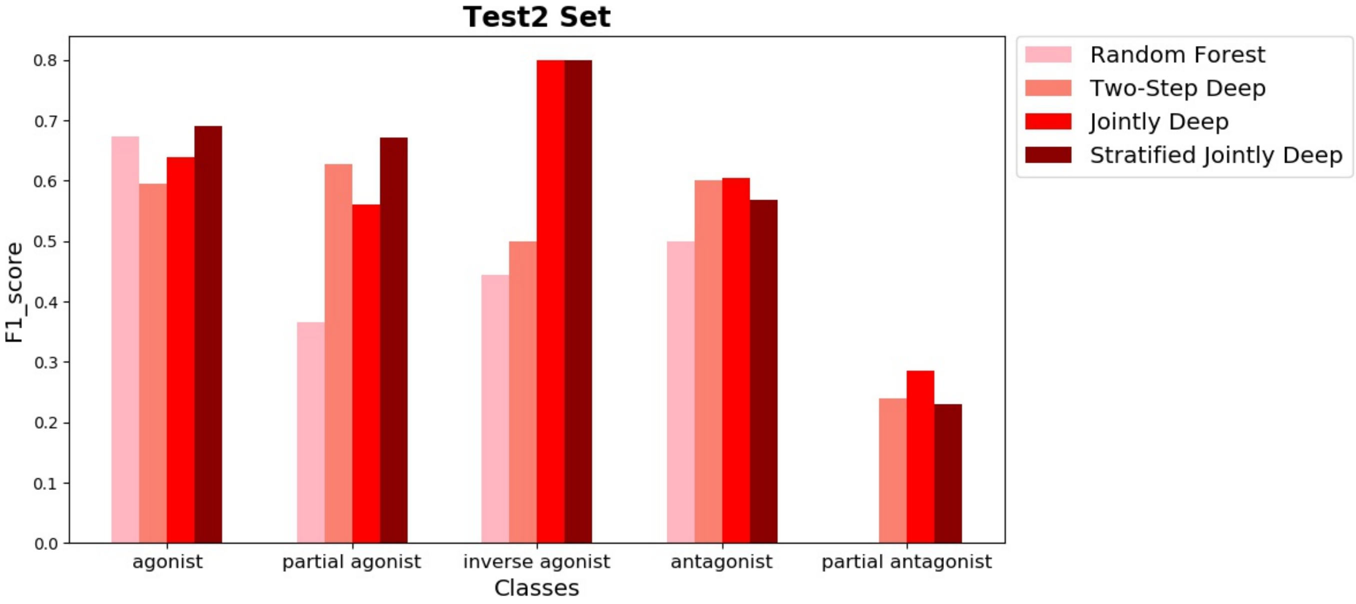

Figure 4.4: F1 score of the Test2 set for the four test sets among four kinds of models

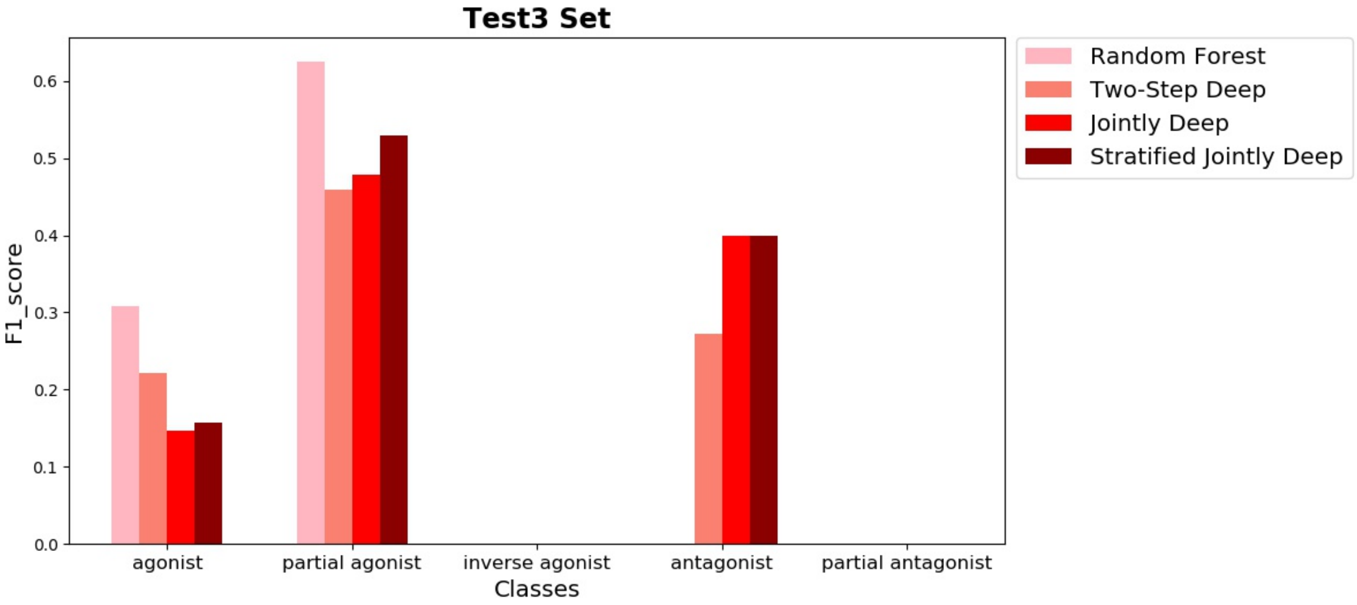

Figure 4.5: F1 score of the Test3 set for the four test sets among four kinds of models

4.1.3 Conclusions

(1) For Test1, Test2 and Test3, stratified jointly trained deep model is the model with the best average F1 score.

(2) The baseline F1 score is 0.2 and it can be told that the average scores of the deep models presents a relatively good results than the shallow model.

(3) Based on the average F1 score, it can be concluded that using deep model instead of the shallow model improves the generalizability to new proteins and new ligands.

(4) Based on the average F1 score, the training ways of the deep model, two-step and jointly training, do not make much difference for the performance in this task.

(5) Based on the average F1 score, the stratified model improve the performance of the prediction. It can be concluded that this method improves the generalizability of the model from the training set to test set. This tells that when training a machine learning model, feature distribution of the data for the test sets need to be taken into consideration. This can be regarded as a semi-supervised problem.

4.2 Results of the transfer learning

4.2.1 F1 score of different training sets

Here are the F1 score for the different training sets. Train origin stands for the result without fine-tuning the model. Train with percentage stands for the results with fine-tuning the model. Different percentage of the training set is used in an accumulating way.

To see the improvement in a more direct way, a line graph was plotted as follows:

Data set	Agonist	Antagonist
Train origin	0.4	0
Train_6%	0.7743	0.4873
Train_10%	0.8085	0.4933
Train_20%	0.8198	0.4376
Train_30%	0.8239	0.4348
Train_40%l	0.8211	0.4383
Train_50%l	0.8170	0.4048

Table 4.5: Joint-trained deep vs Stratified Joint-trained deep

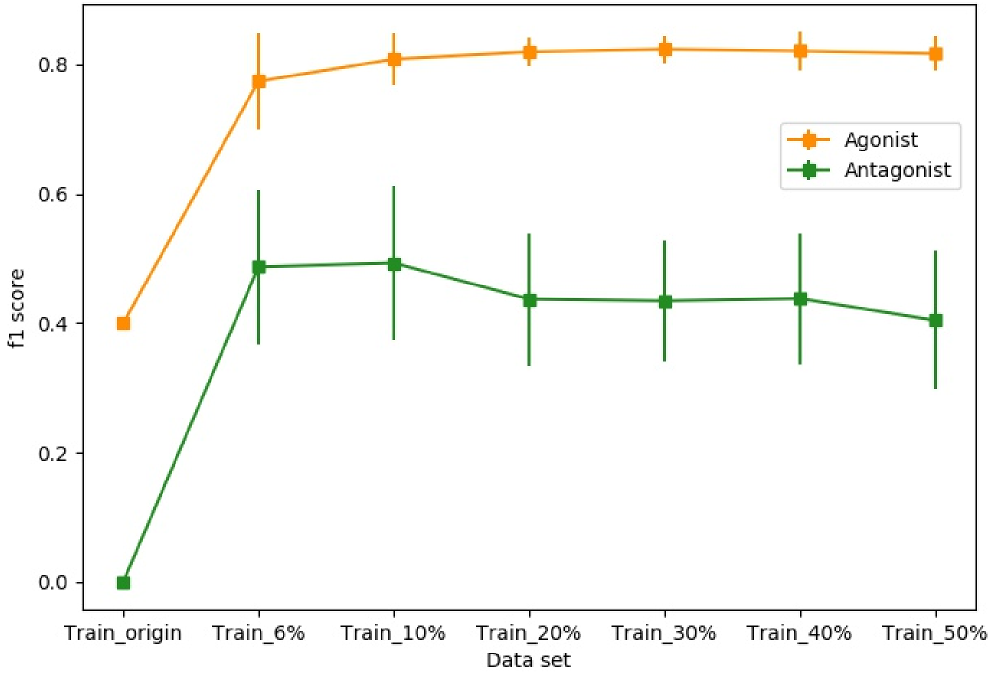

Figure 4.6: F1 score of the orphan test set for the training sets

4.2.2 Conclusions

(1) An obvious rise can be seen for both agonist and antagonist when fine tuning is used for the model. The transfer learning strategy improved the prediction performance of the model.

(2) The predictions for the agonist have an overall better performance than the antagonist. This can be explained that the Rev-erbA-alpha only have agonist compounds.

(3) The similar values of the F1 scores of different percentage of the training sets can be explained by the similar backbones of the compounds. Thus, the increasing data points cannot improve the predictions.

REFERENCES

[1] Wikipedia contributors, “Inverse agonist — Wikipedia, the free encyclopedia,” 2020. [Online; accessed 13-March-2020].

[2] N. Lagarde, N. Ben Nasr, A. Jeremie, H. Guillemain, V. Laville, T. Labib, J.-F. Zagury, and M. Montes, “Nrlist bdb, the manually curated nuclear receptors ligands and structures benchmarking database,” *Journal of medicinal chemistry*, vol. 57, no. 7, pp. 3117–3125, 2014.

[3] M. Reau, N. Lagarde, J.-F. Zagury, and M. Montes, “Nuclear receptors database including negative data (nr-dbind): A database dedicated to nuclear receptors binding data including negative data and pharmacological profile: Miniperspective,” *Journal of Medicinal Chemistry*, vol. 62, no. 6, pp. 2894–2904, 2018.

[4] R. Nanduri, I. Bhutani, A. K. Somavarapu, S. Mahajan, R. Parkesh, and P. Gupta, “Onrldb—manually curated database of experimentally validated ligands for orphan nuclear receptors: insights into new drug discovery,” *Database*, vol. 2015, 2015.

[5] M. Karimi, D. Wu, Z. Wang, and Y. Shen, “Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks,” *Bioinformatics*, vol. 35, no. 18, pp. 3329–3338, 2019.

[6] S. Wang, W. Li, S. Liu, and J. Xu, “Raptorx-property: a web server for protein structure property prediction,” *Nucleic acids research*, vol. 44, no. W1, pp. W430–W435, 2016.

[7] C. N. Magnan and P. Baldi, “Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity,” *Bioinformatics*, vol. 30, no. 18, pp. 2592–2597, 2014.

[8] D. Weininger, “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules,” *Journal of chemical information and computer sciences*,

vol. 28, no. 1, pp. 31–36, 1988.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[10] M.-T. Luong, E. Brevdo, and R. Zhao, "Neural machine translation (seq2seq) tutorial," *https://github. com/tensorflow/nmt*, 2017.

[11] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[13] N. Lallous, S. V. Volik, S. Awrey, E. Leblanc, R. Tse, J. Murillo, K. Singh, A. A. Azad, A. W. Wyatt, S. LeBihan, *et al.*, "Functional analysis of androgen receptor mutations that confer anti-androgen resistance identified in circulating cell-free dna from prostate cancer patients," *Genome biology*, vol. 17, no. 1, p. 10, 2016.

[14] M. Tucci, C. Zichi, C. Buttigliero, F. Vignani, G. V. Scagliotti, and M. Di Maio, "Enzalutamide-resistant castration-resistant prostate cancer: challenges and solutions," *OncoTargets and therapy*, vol. 11, p. 7353, 2018.