Drug-Target Binding Affinity Prediction: Evaluating the Impact of Dataset Size on Model Performance

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Data Analytics
June 2025
Himashi Karunarathna

UNIVERSITY OF TURKU
Department of Computing

HIMASHI KARUNARATHNA: Drug-Target Binding Affinity Prediction: Evaluating the Impact of Dataset Size on Model Performance

Master of Science (Tech) Thesis, 52 p.
Data Analytics
June 2025

Drug-target interaction (DTI) prediction plays a crucial role in modern drug discovery. However, real-world DTI prediction is often challenged due to limited availability of known interaction data on drugs and targets. This study investigates how the size of a training dataset influences the performance of a machine learning model, in particular, Kronecker Regularized Least Square model evaluated using Concordance Index under four experimental settings that represent the realistic scenarios in DTI prediction. These four settings are based on whether the test set contains entirely novel drugs and targets, share either drugs or targets in the training set or includes drugs and targets that are both present during the training phase.

The results of the experiment revealed that the model performance heavily varies depending on the experimental setting. Although the model showed improvement of prediction performance under each setting with increased training data size, some settings showed much improvement and better generalization ability than the others. As expected, the model performed best when the model has seen interactions involving the drugs and targets in the test set during training. The model performed well when predicting interactions for unseen targets but struggled to generalize well with novel drugs or both novel drugs and targets.

The findings of the study highlight the importance of addressing each experimental setting individually. While some experimental settings benefited significantly well with the available training data size, others showed only slight improvement though with an upward trend suggesting that more data may help to further improve the performance.

Keywords: Drug-target interaction, DTI, Binding affinity prediction, KronRLS, Machine Learning, Evaluation

Contents

List of Figures

List of Tables

List of Acronyms

AULC	Area Under the Learning Curve
CADD	Computer Aided Drug Design
CI	Concordance Index
DTI	Drug-Target Interaction
GPCR	G-protein Coupled Receptors
HTS	High-Throughput Screening
KronRLS	Kronecker Regularized Least Squares
LBVS	Ligand Based Virtual Screening
ML	Machine Learning
QSAR	Quantitative Structure–Activity Relationship
S1	Experimental Setting 1
S2	Experimental Setting 2
S3	Experimental Setting 3
S4	Experimental Setting 4
SMILES	Simplified Molecular Input Line Entry System
SW	Smith-Waterman
SVM	Support Vector Machine
TC	Tanimoto Coefficient
VS	Virtual Screening

1 Introduction

Traditional drug discovery is an expensive process that can take many years before a single drug reaches the market [1]. On top of the time and cost involved, the failure rate is also extremely high, with many potential drug candidates failing during clinical trials. This high cost, time and risk has made drug discovery a major challenge and researchers are constantly trying to find more productive and efficient solutions.

The use of *in silico* methods(computer aided approaches), for predicting Drug Target Interaction (DTI) has gained much attention over the recent years. These computational methods allow researchers to identify high-potential drug candidates early in the process, reducing the need for high-throughput screening (HTS) which is both expensive and time-consuming. Advances in these computer aided approaches are helping to make drug discovery faster, more cost-effective, and more targeted. Among various *in silico* methods, such as docking simulations [2], [3] and ligand based virtual screening [4], Machine Learning(ML) has become a powerful and promising approach for predicting DTIs.

There are many different ML approaches used for DTI prediction but often the studies tend to overestimate the ability of the prediction models because the experimental settings do not represent the real world scenarios [5]. The study by Pahikkala et al. [5] explains four factors that can highly affect DTI prediction results of an experiment. They are (i) whether the problem formulation is binary or regression,

(ii) The data set used for evaluation, (iii) using simple or nested cross-validation as the evaluation procedure and (iv) experimental settings. The fourth factor explains four experimental settings that are defined based on whether the test set contains entirely novel drugs and targets, share either drugs or targets in the training set or includes drugs and targets that are both present during the training phase. This enables the evaluation of the prediction model under more realistic conditions. The authors of the above study have shown the importance of addressing these settings in the prediction tasks by showing how these different settings affect the accuracies of the models. The four settings are,

Setting S1: The model predicts interactions for drugs and targets that were both present during the training phase. This represents the simplest case.

Setting S2: The model predicts interactions for new drugs, but the targets are already seen by the model during the training phase.

Setting S3: The model predicts interactions for new targets, but the drugs are already seen by the model during the training phase.

Setting S4: The model predicts for entirely new drugs and targets which were not present during the training phase. This is the most realistic as well as the most challenging scenario.

All the therapeutic drugs that are used today, belong to the Chemical Space [6] which is the set of all possible molecular structures that can exist theoretically. The size of the chemical space is enormous and also depends on how the space is defined. Therefore, one can also argue that it is infinite. The known set of molecules is only a small fraction of this space and the fraction explored for medicinal purposes is even smaller [7]. Databases such as Pubchem contains a huge amount of chemical structures of compounds and protein structure information. However, the information available on the interactions between these chemicals and proteins are relatively limited [8]. This is a major challenge faced by computational DTI

prediction methods. Finding and measuring interactions using traditional methods is costly in terms of both time and money. Therefore, it is important to not only find methods that are capable of leveraging this existing data but also study how size of the training dataset impacts on the performance of prediction models.

The setting S1 represents the scenario where the interactions involving drugs and targets in the test set are available to the model. In settings S2 and S3, model only see interaction data on either drugs or targets in the test set whereas in setting S4, no prior knowledge on interactions involving test drugs and targets is available. ML models are known to perform well with drug target interaction prediction. When it comes to predicting interactions for unknown drugs and targets, it seems that the models find it somewhat difficult to learn patterns [5] particularly in the setting S4, where the model is expected to predict interaction affinities for unknown drugs and targets. It is important to treat these problem settings differently and understand the impact of data size on the performance to determine future steps in DTI prediction.

1.1 Research Question

This study investigates the following research question.

> How does the data size of a training dataset impact the performance of a drug-target interaction binding affinity prediction model under the above mentioned experimental settings S1-S4.

1.2 Objectives

1. Investigate the impact of training dataset size on overall model performance in each experimental settings S1–S4.

2. Analyse learning curves plot against the training dataset size in each setting.

3. Evaluate the model's generalization capability under each experimental setting.

4. Compare the prediction performance of each experimental setting.

Apart from the lack of representation of the four experimental settings in ML based DTI prediction, most research has been carried out treating DTI interaction prediction as a classification problem. When compared with binding affinity prediction which is a regression task, the classification models show higher performance as the classification settings simplifies the problem. Although the binary classification approach has its own applications, the regression based binding affinity prediction gives more informative insights as it measures the binding strength. This study will help to determine the model learning capabilities and the data requirements focusing not only on different experimental settings but also on treating DTI prediction as a regression problem.

This thesis document is organized as follows. Chapter 1 provides an introduction to the thesis by presenting the motivation behind the research, research question and the objectives that guide this experiment. Chapter 2 provides background information on the foundational concepts and methodologies that are relevant to this work. A review of related work in the area of DTI prediction with a focus on machine learning techniques is presented in Chapter 3. Chapter 4 describes the data used in this study including the drug-target binding affinities, drug-drug similarities and target-target similarities. Chapter 5 provides specifics on the selected machine learning model and a detailed explanation on how the data is partitioned according to the four experimental settings. This is followed by Chapter 6, which presents the results obtained from each experimental setting, organized according to the stated research objectives and Chapter 7 which discusses the findings focusing on each

experimental setting. Finally, Chapter 8 concludes the thesis by summarizing the contributions and limitations of the study along with suggestions for future work.

2 Background

2.1 Machine Learning

ML is a field of study that focuses on the development of algorithms capable of learning from data without the need for explicit programming. These learning algorithms can be defined as a set of instructions which enables the machines to identify relationships and complex patterns hidden in data that can be difficult for humans to detect. Unlike traditional programming techniques, where computers are given step by step instructions by developers, ML systems learn such rules from data and has the ability to generalize and make accurate predictions to new inputs based on the learned patterns in data.

ML is often used in applications where the traditional approaches are inadequate, either because the problem is too complex or because it requires a huge set of rules and extensive fine tuning. ML can be well suited when the data it learns from keeps changing over time or when the goal is to extract insights from large amounts of data [9]. ML is used in a wide variety of applications across various fields and has gained a significant amount of attention for its potential in predicting drug-target interactions over the years.

Machine learning techniques are typically grouped into three main types.

- Supervised Learning - Learn from labeled data and predict for new, unseen inputs

- Unsupervised Learning - Discover patterns in unlabeled data
- Reinforcement Learning - Learns by interacting with an environment and adjusting based on feedback (rewards or penalties)

This thesis focuses exclusively on supervised learning.

2.2 Supervised Machine Learning

Supervised training involves training machine learning models using labeled data where z_i represent each data point consist of a pair of input $x_i \in X$ and its corresponding output $y_i \in Y$. X and Y defines the input and output spaces which represents all the possible values for x and y. Z is the Cartesian product of the input and output spaces. It represents the space of all possible input-output pairs. A training sample S consists of n number of data points,

$$S = (z_1, \cdots, z_n) \in Z^n \qquad Z = X \times Y$$

$$z_i = (x_i, y_i) \quad \text{where} \quad x_i \in X \quad \text{and} \quad y_i \in Y$$

In supervised learning, the objective is to learn a function $h : X \to Y$ that can predict the correct output y for a given input x. h is referred to as a hypothesis. In the learning process, the algorithm considers all the possible hypotheses which is known as the hypothesis space $\mathcal{H}$.

Supervised learning can be categorized into two main types as classification and regression based on the nature of the output variable the model is trained to predict.

- Classification - The model predicts a discrete value either binary $y_i \in \{-1, 1\}$ or a value from a predefined set $C = \{c_1, \cdots, c_k\}$
- Regression - The model predicts continuous values $y_i \in \mathbb{R}$

In DTI prediction, both classification and regression tasks can be found in the literature as some studies treat drug-target interaction as a binary problem while others focus on predicting the strength of bindings. However, this thesis focuses on predicting the binding affinities between compounds and proteins which is a regression based task.

Learning algorithm $\mathcal{A}$ can be defined as,

$$\mathcal{A} : \bigcup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{Y}^{\mathcal{X}}$$

Here $\mathbb{N}$ represents the set of natural numbers indicating that the algorithm takes a finite set of labeled training examples $(x_i, y_i) \in X \times Y$ as inputs and produces a function (a hypothesis) that maps inputs to outputs.

The learning algorithm is responsible for selecting a suitable hypothesis from the hypothesis space $\mathcal{H}$. The goal is to identify the hypothesis which minimizes the objective function which measures the error between the predicted and the actual outputs. The optimal hypothesis not only fits well with the training data but also has the ability to generalize to unseen data which is a fundamental aspect of machine learning. If a model is too complex, there is high variance and it will fit the training data very closely giving low training error but poor generalization. This is called overfitting. If the model is too simple with high bias then it may fail to capture any meaningful pattern. This is called as underfitting. The optimal solution lies in between these two and it is often described as the bias-variance trade-off.

2.3 Performance Evaluation

Performance evaluation is a critical component of machine learning as it plays a significant role when developing high quality accurate models. It is not only used to estimate performance of a model on novel data but also to compare and assess

models trained with different algorithms and hyperparameters, in order to select the most suitable approach for a given task [10]. A good evaluation process helps to determine that a model with high performance is capable of generalizing well to unseen data rather than overfitting to its training dataset.

The size of a dataset can affect the reliability and stability of a performance estimation. In smaller datasets a specific split between training and testing sets can result a high variance in performance metrics. This is important to consider in this study, as we investigate how the model performance varies with the dataset size by systematically dividing it into smaller subsets.

2.3.1 Hold Out Method

This study is focused on estimating the performance of the model using the hold out method where the dataset is randomly divided into training set $\mathcal{S}_{\text{train}}$ and test set $\mathcal{S}_{\text{test}}$ such that $\mathcal{S}_{\text{train}} \cup \mathcal{S}_{\text{test}} = \mathcal{S}$ and $\mathcal{S}_{\text{train}} \cap \mathcal{S}_{\text{test}} = \emptyset$.

The performance evaluation measure m on algorithm $\mathcal{A}$ using a hold out method $\mathcal{HO}$, can be expressed as the prediction performance of the selected hypothesis h evaluated on the test set.

$$m_A^{HO}(\mathcal{S}) = m_h(\mathcal{S}_{\text{test}})$$

When it comes to performance evaluation, close attention must be given to maintain a clean separation between the train and test data. Otherwise, it could lead to data leakage resulting an overestimated model performance. Apart from not having a clean separation, data leakage can also occur due to practices such as data pre-processing or feature selection with both train and test sets [11]. Such practices can expose information about the test data to the model before the evaluation step. However, the model might not be able to reproduce the same performance with new data.

In this study, to analyse and compare the learning behavior of the model with

increasing data size across different settings, we need to maintain a fixed test set. Although cross validation [10] is commonly used for performance estimation on small datasets as it ensures all data points contribute to both training and validation across multiple folds, it is not suitable for this study as it does not maintain a fixed test set.

2.3.2 Repeated Hold Out Method

While hold out method is straight forward and widely used, its performance estimations can suffer from high variance, especially when the dataset is small and a single train-test split may not be representative. To mitigate this issue, repeated hold out validation can be used which involves performing hold out method multiple times with different random splits.

If the evaluation is done k times with different random seeds, the average performance metric $\text{Metric}_{\text{avg}}$ is computed as follows where Metric_j is the performance of the model on the j^{th} test set.

$$\text{Metric}_{\text{avg}} = \frac{1}{k} \sum_{j=1}^{k} \text{Metric}_j$$

Repeated hold out approach reduces the variance that can result from a single train-test split.[10]. As the model must be trained and evaluated k times, the computational cost is much higher. However, if hold out method is used, repeated approach is necessary particularly in scenarios where the performance estimations are sensitive to data partitioning such as small or imbalanced datasets.

2.3.3 Evaluation Metrics

Depending on the problem formulation, different evaluation metrics are used in machine learning. For example in classification tasks, metrics such as accuracy,

precision and recall are used. Metrics such as Mean Squared Error(MSE) and Mean Absolute Error(MAE) are widely used to assess the performance of regression tasks.

In this study, Concordance index also known as the C-Index(CI) is used as the evaluation metric. Rather than predicting the exact value of the drug target binding affinity, the goal is to determine which drug-target pairs are having good binding strength over others. CI is a rank based method which means that it is concerned about the order of the predictions. In the topic of DTI prediction task with binding affinities, CI measures the likelihood that for two randomly selected drug-target pairs, the model predicts binding affinities preserving the order of their actual values.

If the model predicts f_i and f_j for actual values y_i and y_j where $y_i > y_j$, it checks whether $f_i > f_j$.

$$\text{CI} = \frac{1}{Z} \sum_{y_i > y_j} h(f_i - f_j)$$

Z is the normalization constant which represents the total number of valid comparisons and the function h assigns a score based on the comparison of prediction pairs.

$$h(f_i, f_j) = \begin{cases} 1, & \text{if } f_i > f_j \\ 0.5, & \text{if } f_i = f_j \\ 0, & \text{if } f_i < f_j \end{cases}$$

The CI is calculated using the method described in Algorithm 1 which takes a list of true outcome values and a list of predicted values.

The method initializes counters for comparable pairs(n) and concordance pairs (h_num). Two pairs are considered as comparable if the true labels of the two pairs are different. The method iterates through all unique pairs indexed by i and j where $j > i$ to avoid duplicates and identifies the comparable pairs. n is incremented by 1

for each comparable pair.

A pair is considered concordant when the order of the predicted values are in the same order as the actual values. For each comparable pair, if the pair is concordant with each other then (h_num) is incremented by 1. if predicted scores are equal, (h_num) increment by 0.5 whereas if pairs are discordant they are not counted. Final CI is calculated as,

$$\text{CI} = \frac{\texttt{h\_num}}{n}$$

Algorithm 1: Compute Concordance Index

Input: List of true outcome values `true_labels`, list of predicted values `pred_labels`

Output: Concordance index `c_index`, a float between 0 and 1

```
num_comparable_pairs ← 0 ;    // Total number of comparable pairs
num_concordant ← 0 ;                  // Number of concordant pairs
for i ← 0 to length(true_labels) − 1 do
    true_i ← true_labels[i] ;
    pred_i ← pred_labels[i] ;
    for j ← i + 1 to length(true_labels) − 1 do
        true_j ← true_labels[j] ;
        pred_j ← pred_labels[j] ;
        if true_i ≠ true_j then
            num_comparable_pairs ← num_comparable_pairs +1 ;
            if (pred_i < pred_j and true_i < true_j) or (pred_i >
             pred_j and true_i > true_j) then
                num_concordant ← num_concordant +1 ;    // Concordant
                 pair
            end
            else if pred_i == pred_j then
                num_concordant ← num_concordant +0.5 ;       // Tie in
                 prediction
            end
        end
    end
end
c_index ← num_concordant / num_comparable_pairs ;
return c_index
```

Calculated CI value can be interpreted as follows.

- CI = 1 → A perfect prediction where all pairs are in order
- CI = 0.5 → A random prediction. The model is not learning any useful patterns.
- CI < 0.5 → Worse than random. Inversely predicting the actual outcome

2.3.4 Learning Curves

Learning curves are beneficial in machine learning, particularly in guiding decisions related to data collection, early stopping of model training, and selecting the most appropriate model [12] [13].

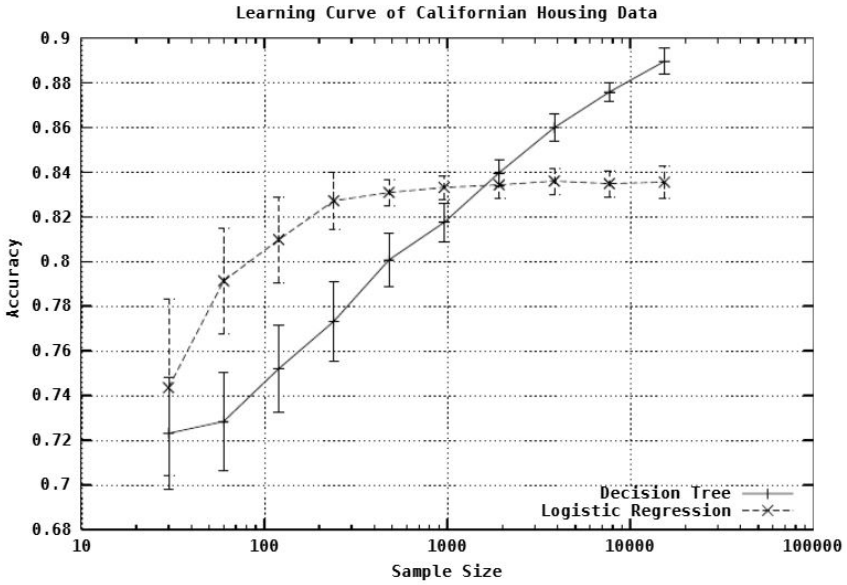

Figure 2.1: Comparing learning curves of two machine learning approaches [14]

Learning curves are useful to assess how efficiently models, like KronRLS in our study, utilize available data. They also support comparative analysis between models (Figure 2.1) by illustrating which models continue to improve as the data size increases and which tend to plateau early potentially due to the limitations in

the model capacity or data representations . In this study, instead of comparing different model architectures, we focus on comparing the learning behavior across the four experimental settings.

Learning curves are important diagnostic tools which can offer insights on learning capability and the generalization of models. Some studies plot curves using error or loss. However, in this study we will be using c-index plotted against the training dataset size. We are interested in the shape of the curves as well as the gap between the training and testing curves. A larger gap while having a higher training performance and a lower testing performance indicates a problem of overfitting. If both curves are showing poor performance, it can mean a problem of underfitting where the model is facing difficulty to find meaningful patterns in data. By looking at the learning curves, we can get insights on whether the model will improve with more data. When the curves have reached a plateau more data will not help to improve the performance. In such cases, hyperparameter tuning, alternative learning algorithms or different data representations can be helpful.

To compliment visual inspections, we compute the Area Under the Learning Curve (AULC) [13] which is a metric that is often used to summarize a model's performance with varying data sizes. However, it is important to interpret this value as different learning behaviors can result the same value. Therefore, AULC should be used alongside the shape of a learning curve to gain a complete understanding of model's efficiency.

2.4 Pair Input Data

In pair input data, each input x is represented as a pair of objects. In drug target interaction this can be viewed as a drug target pair (d_i, t_i).The trained model is expected to predict the drug target binding affinity $y = f(d_i, t_i)$ for new pairs where $y \in \mathbb{R}$ [15]. Unlike in regular inputs, two pair input observations may share their

objects introducing dependencies between them. These dependencies should be taken into consideration when evaluating model performance.

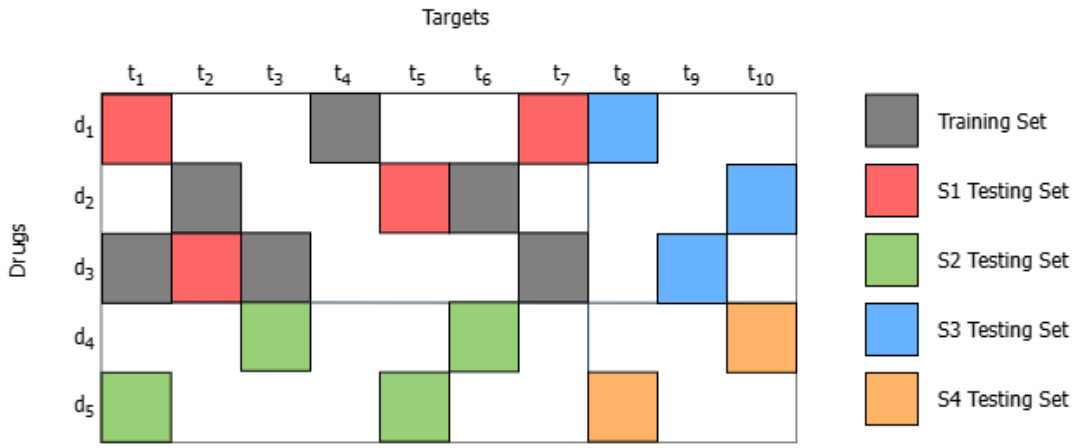

Figure 2.2: Illustration of pairwise drug-target settings. How training and testing sets differ from each setting [15]

The drug space and the target space are denoted by D and T. The drug-target space which includes the complete set of possible drug target pairs is given by $X = D \times T$. Given a dataset with drug-target pairs Z_{obs} where $D_{\text{obs}} \subset D$ and $T_{\text{obs}} \subset T$ are the unique drugs and targets observed in the dataset. When the model try to predict the binding affinity for a new drug-target pair (d_i, t_i) following 4 scenarios should be considered,

1. known drug and known target: $d \in D_{\text{obs}}$ and $t \in T_{\text{obs}}$
2. new drug and known target: $d \notin D_{\text{obs}}$ and $t \in T_{\text{obs}}$
3. known drug and new target: $d \in D_{\text{obs}}$ and $t \notin T_{\text{obs}}$
4. new drug and new target: $d \notin D_{\text{obs}}$ and $t \notin T_{\text{obs}}$

2.5 Kernel Methods

While many traditional machine learning methods excels at modeling linear relationships, real world data often exhibits complex non-linear patterns. Kernel methods bridge this gap through 'kernel trick' which implicitly transform input data into high dimensional feature spaces where data may become linearly separable. By avoiding explicit computations of high dimensional feature space and solely relying on kernel evaluations, this maintains computational efficiency while learning complex patterns [16].

Kernel Methods rely on functions known as kernels which compute similarity between two data points. Kernel functions produce kernel matrices. These matrices satisfy two key properties.

1. Symmetric: $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$

2. Positive Semi-Definite (PSD): Ensure that the kernel corresponds to a valid inner product in some feature space.

To be used in a kernel-based machine learning model a similarity matrix should pass the above two conditions. In DTI prediction problems, kernel methods are particularly valuable as they can incorporate chemical and genomic prior knowledge in the form of drug-drug and target-target similarity matrices. It is also convenient for some complex data structures be represented as similarity matrices rather than feature vectors. The matrix used to represent n number of objects will always be $n \times n$.

2.6 Kronecker Regularized Least Squares (KronRLS)

In classical least squares method, the goal is to find the function which minimizes the squared error between the predicted and true outputs.

$$\sum_{i=1}^{n}(y_i - f(x_i))^2$$

However, this method can overfit to the training data, especially in scenarios where there are high dimensions or when data contain outliers. This overfitting might lead to complex models. To address this issue, the regularization term is added. The two most common approaches are,

1. Lasso Regression (L1 Regularization) - Use the absolute value of coefficients as the penalty term

2. Ridge Regression (L2 Regularization) - Use the square of coefficients as the penalty term

KronRLS method is build upon ridge regression.

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\|f\|_k^2$$

Here, λ is referred to as the regularization parameter and $\|f\|_k^2$ denotes the squared norm of the function f in the Reproducing Kernel Hilbert Space (RKHS) associated with the kernel k. This term penalizes model complexity to improve generalization. A small λ allows the model to fit to training data which may lead to overfitting and a higher λ increases the weights of the term leading to a more simpler model. Therefore, the regularization term mused be tuned to find the balance between these two.

In this study we use KronRLS provided by the RLScore [17] library which is a open source Python library designed for kernel based machine learning. It offers

various implementations of regularized least squares algorithms and also supports multiple kernels including pre-computed externally created kernels. KronRLS is specifically designed for pairwise input data such as drug-target binding affinities $x_i = (d_i, t_i)$ where d_i is a drug and t_i is a target.

Let us denote $K_D \in \mathbb{R}^{n_D \times n_D}$ as the kernel of drugs and $K_T \in \mathbb{R}^{n_T \times n_T}$ as the kernal of targets. The Kronecker product which can be viewed as the interaction between all drug-target pairs can be represented as,

$$K\big((d_i, t_i), (d_j, t_j)\big) = K_D(d_i, d_j) \cdot K_T(t_i, t_j)$$

The Kronecker product allows KronRLS to model interactions between two domains. However, forming the full Kronecker product matrix is computationally expensive. KronRLS avoids this issue using optimization techniques.

3 Related Work

3.1 DTI Prediction

There are three main approaches used to study Drug-Target Interactions(DTI). *In vivo* approaches represent experiments conducted on living organisms such as human clinical trials. *In vitro* experiments are tests done outside living organism typically in controlled laboratory settings while *in silico* approaches involve methods carried out with the aid of computer programs [18]. Rather than relying on a single method, DTI studies follows a step by step approach utilizing these different approaches.

In vitro experiments are widely recognized for their efficiency and reliability in studying drug-target interactions, particularly with the use of HTS setups. However, these experimental approaches are often time-consuming and costly [19]. To address these challenges, computer-aided drug design (CADD) approaches such as virtual screening (VS) have emerged as powerful alternatives, significantly shortening the time and lowering the cost involved in drug development. While virtual screening methods like ligand-based virtual screening (LBVS) [4] have proven valuable for predicting drug candidates it has significant limitations as it heavily relies on known active ligands in a target protein of interest. Structure based VS methods such as molecular docking [2] rely on the 3D structure of target proteins for DTI prediction. As a result, it cannot be applied when the 3D structure of a target is unavailable which is often the case for membrane proteins, such as GPCRs (G-protein coupled

receptors) [20].

To mitigate these limitations more research was focused on methods that uses chemogenomic approaches [8]. These methods integrate information from both chemical space with chemical structure information and genomic space with target protein sequences to predict DTIs with currently available drug target interaction data. Early DTI research focused on one drug-one target approach. However, multiple studies has revealed that drug-target interactions are far more complex and diverse [21] [22] [1]. Predicting new drugs for established targets and discovering new treatment possibilities for the existing drugs [20] can be identified as key features in the drug-target interaction.

DTI prediction is not only useful for discovering new drugs but also for drug side effect prediction [23] and drug repurposing [24], defined as the process of discovering new therapeutic applications for established drugs beyond their original indications. This can be especially valuable in situations where developing a new drug from scratch might not be practical.

3.2 Machine Learning in DTI Prediction

Numerous ML approaches are used to predict drug-target interactions. Supervised machine learning methods in DTI prediction can be categorized into two main branches as feature vector based methods and similarity based methods [25].

In feature vector based methods data is represented as feature vectors that are generated combining chemical descriptors of drugs and sequence of targets and standard machine learning models for the prediction [20] [26].

Similarity/Distance-based functions [27] use chemical structure similarities between compounds and sequence similarities between proteins to predict DTIs using methods such as nearest profile or bipartite graph learning. The underlying concept of similarity based methods is that two compounds with high structural similar-

ity are likely to interact with target proteins that has high sequence similarity and likewise proteins that are close in the network tend to interact with similar drugs [24]. The study by Yamanishi et al. [27] used a similarity based approach with four different drug–target classes: GPCRs, ion channels, enzymes and nuclear receptors. Importantly, this dataset included membrane proteins such as GPCRs and ion channels which are challenging for structure based prediction methods such as docking simulations due to the lack of 3D structures. The data sets introduced by this study was later used as the gold standard in early DTI prediction works.

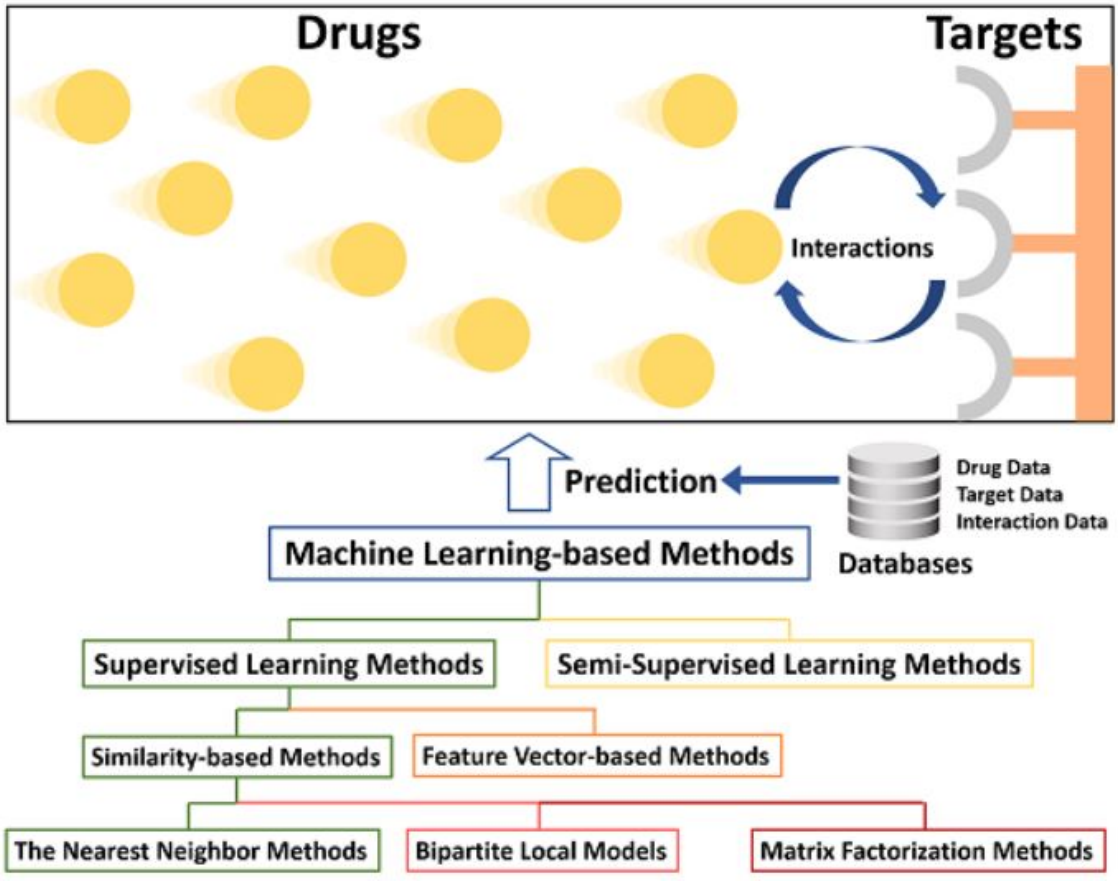

Figure 3.1: Machine learning approaches used in drug-target interaction prediction [25]

Another recent survey conducted by Bagherian et al. [28], has systematically categorized machine learning approaches used in the DTI prediction into six branches. In addition to the above mentioned feature based models and similarity based ap-

proaches, the authors identified matrix factorization [29], network based methods [30], hybrid methods and deep learning techniques.

The survey paper by Bagherian et al. [28] also states the issue of many data sets used for ML based research in DTI being in binary nature [27] [31] and the importance of having continues values for DTI binding affinities as it will be more useful and meaningful because binding affinities between drug target pairs naturally vary across a spectrum. The 'gold standard' data sets [27] used in the early DTI prediction experiments also contains binary data. This issue was also mentioned in the paper by Pahikkala et al. [5] suggesting to use the kinase binding assay data from Davis et al. [32] as a bench marking data set in the future experiments due to continues values and high quality.

Recent advances in drug-target interaction prediction has adopted deep learning approaches with different architectures such as DeepDTA [33], WideDTA [34] and GraphDTA [35] which are focused on predicting drug target binding affinities. While these studies have demonstrated strong performance, they do not address the challenge of predicting the interactions that involve unknown drugs and targets.

3.2.1 Kernel Methods in DTI Prediction

Kernel methods are applicable to the four experimental settings in pairwise learning and are suitable for a wide range of application domains including DTI prediction. Although some other approaches may achieve higher performance in certain domains, they often do not generalize well across all four settings [36]. Biological entities such as protein structures are easier to represent using their similarities rather than using feature vectors. There are several types of kernel functions often used in pairwise learning such as linear, polynomial and Gaussian also known as RBF. A comprehensive overview of most commonly used pairwise kernels focusing on improving computational efficiency, can be found in the study by Viljanen et al.

[15].

Different experiments use different types of data to construct kernel matrices [31] [37]. The study by Van Laarhoven et al. [31] has used Gaussian kernels to build kernels from binary vectors called interaction profiles. In this case, a single vector for a drug, represents whether an interaction is present or absent for all the targets in that network which is the target interaction profile for that particular drug. The primary assumption under this study is if two drugs interact in a similar manner with targets in a known drug target interaction network, they will likely interact similarly with a novel target. The study has compared performance of several kernel based methods given different kernels as inputs. These include kernels constructed with drug-target interaction network topology information, chemical and genomic sequence information and a combined version of the above two types. Another study proposed using target kernels based on biological hierarchies to predict interactions between proteins and small molecules [38].

Support Vector Machine (SVM) [20] and KronRLS [5] are widely used kernel-based pairwise learning algorithms. Apart from DTI, the study by Ben-Hur and Noble [39] has used kernel based methods for protein-protein interaction prediction where it compares the performance of a SVM classifier with a diverse set of kernels.

4 Data

In DTI prediction, many methods such as KronRLS and SimBoost [40] use a combination of interaction data and side information like drug-drug and target-target similarities to enhance model generalizability especially for unseen drug or target entities. At the same time, there are also recent deep learning methods that learn representations from raw data such as protein sequences or SMILES (Simplified Molecular Input Line Entry System) without relying on predefined similarity scores [33].

4.1 Drug-Target Binding Affinities

In this study, we use the dataset Davis [32] which contains dissociation constant(K_d) values that quantify the binding affinity between kinase inhibitors and their targets. The dataset contains K_d values for 68 unique kinase inhibitors(drugs) and 442 kinases(targets) which covers more than 80% of human catalytic protein kinome.

The Davis dataset is considered a full dataset as it provide binding affinities for all the 30056 drug-target pairs in the dataset. Lower K_d values indicate strong binding while higher K_d values indicates weaker binding. The distribution of Davis dataset is highly skewed as illustrated in figure 4.1. Majority of the data pairs are having no binding or very weak binding in the dataset.

This experiment also uses drug-drug similarity and target-target similarity data sets from the study by Pahikkala et al. [5].

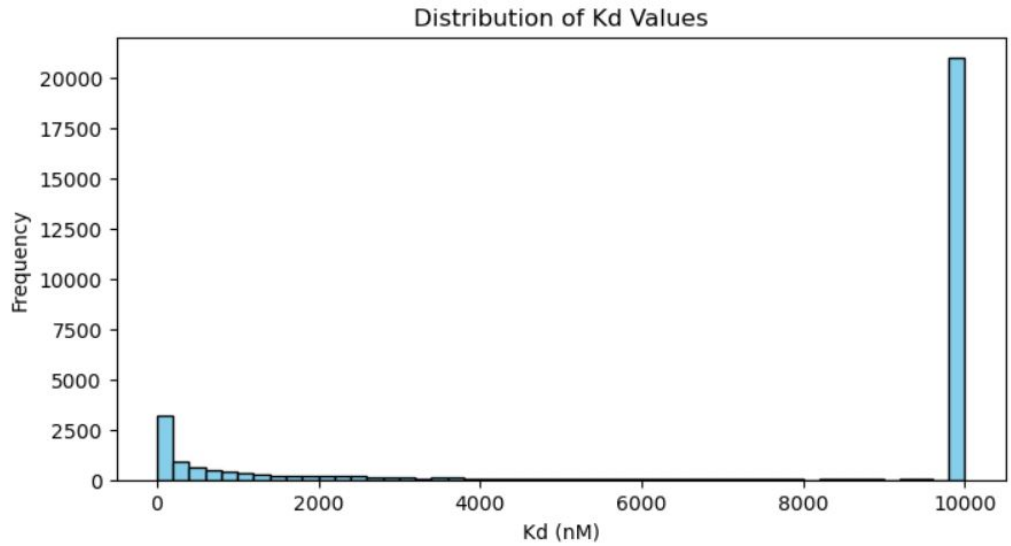

Figure 4.1: Data distribution of binding affinities

4.2 Drug-Drug Similarities

The drug-drug similarity dataset was calculated using Tanimoto Coefficient(TC) [41] that compares molecular fingerprints which are binary representations of structural features of drugs (figure 4.2). It ranges from 0 to 1 where 0 means no similarity and 1 is identical.

$$\mathrm{TC}(A, B) = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

Here, A and B are two fingerprints and N_A and N_B are the number of features present in them respectively while N_{AB} represents the number of shared features between them.

The drug-drug similarity dataset shows a normal distribution centered around 0.55 (figure 4.3). This indicates that the majority of drug-drug pairs are showing a moderate structural similarity while highly similar or highly dissimilar pairs seem to be rare.

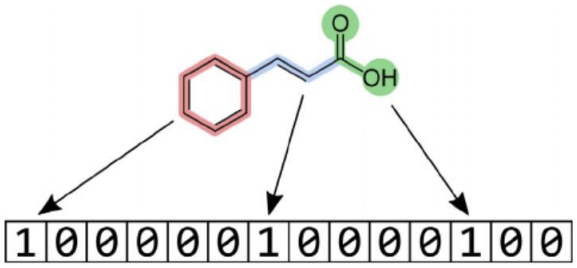

Figure 4.2: Molecule is converted into a fingerprint vector where "1" indicates the presence of specific substructures at a specific position [42]

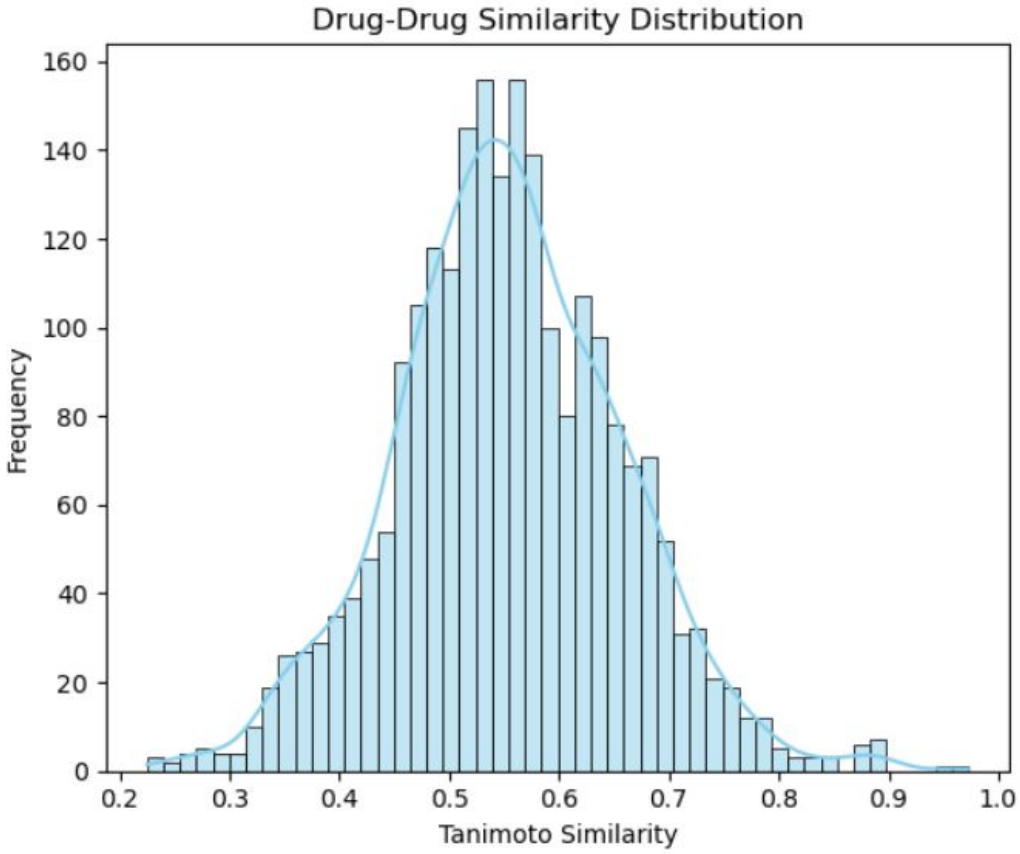

Figure 4.3: Distribution of drug-drug similarities

4.3 Target-Target Similarities

Target-target similarities were calculated using Smith-Waterman(SW) score [43]. SW algorithm is well suited for target sequence comparison as it is designed to handle the complexity and length of biological sequences such as proteins. Higher SW scores indicate that there are more similar regions in the compared sequences.

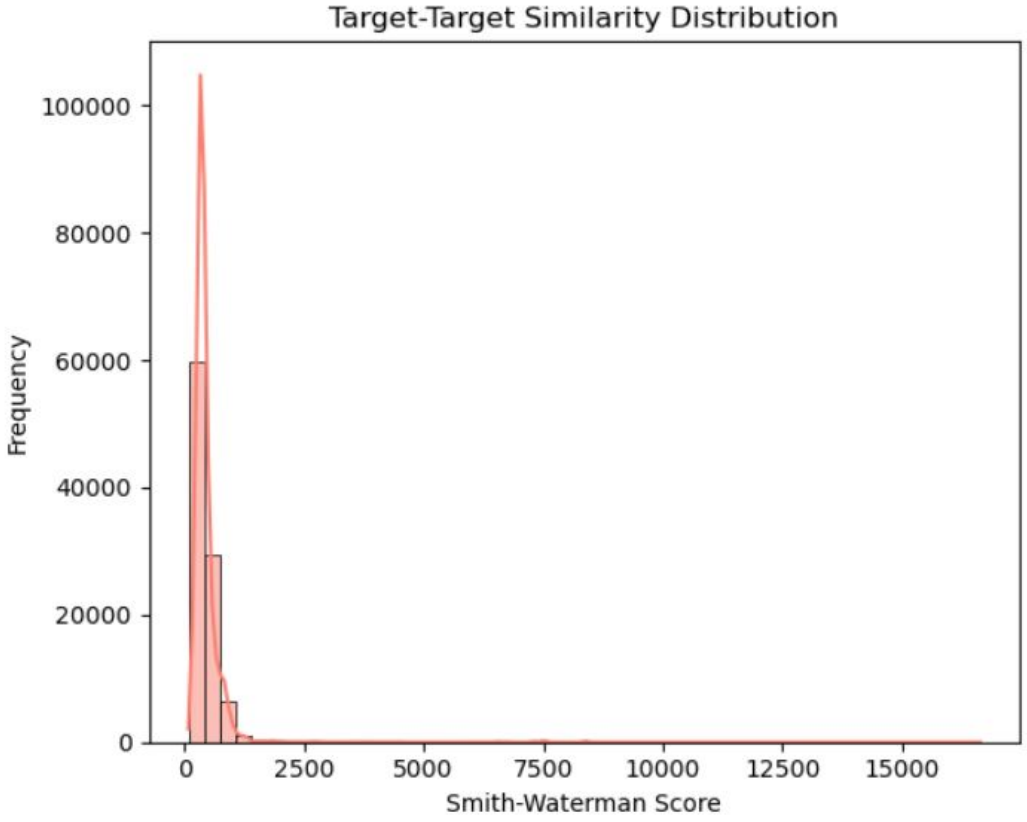

Figure 4.4: Distribution of target-target similarities

Figure 4.4 shows the distribution of the target-target similarities which is heavily skewed towards the lower similarity values. Only a small number of target pairs seem to have high similarities. This indicates that there's a high diversity among the target sequences in the dataset.

5 Experimental Setting

5.1 Model

In this study, we used the Kronecker Regularized Least Squares (KronRLS) method in RLScore [17] to train the DTI binding affinity prediction model.

RLScore facilitates use if multiple kernel functions such as Linear kernel, Gaussian kernel and Polynomial kernel. It also allows the use of pre-computed kernels if needed. In this study we will be using the Linear kernel.

$$k(x_i, x_j) =< x_i, x_j > + \text{bias (default} = 1.0)$$

Even though we are using drug similarity and target similarity matrices as inputs, since they are not positive semi definite kernel matrices, they were used as feature representations. For instance, each row in drug-drug similarity matrix will be considered as a feature vector representing the similarity of one drug to all the other drugs in the dataset.

Using the above kernels and linear KronRLS, a model was trained for each experimental setting. Linear KronRLS learns a weight matrix of size, number of features in $X_1\times$ number of features in X_2 where X_1 and X_2 are feature matrices of drugs and targets.

The initial training data sets created for each setting was split further into smaller parts(refer to 5.2). Models were trained starting from 10% of the whole training set.

Then the number of drug target pairs in the training was gradually increased by 10% in each iteration until the model is trained on the full training data set while keeping record of the c-index in each iteration.

In setting S1 the binding affinity matrix samples fed to the model during the training phase is sparse. The KronRLS model is not capable of handling missing or incomplete data and performing imputation in this context can be misleading. RLScore provides an iterative Kronecker RLS training algorithm called CGKronRLS and in setting S1 we will be using this model.

5.2 Data Partitioning and Sampling

This study adopts the four experimental settings (S1–S4) proposed by Pahikkala et al. [5] to evaluate model performance under different scenarios.

To systematically evaluate how the model performance behave with the number of unique drug-target pairs, four separate experiments were carried out for settings S1-S4. Therefore, for each setting, the data set was divided into initial train and test sets adhering to the conditions of each setting.

The training datasets of each setting were then further divided to 10 equal parts in order to gradually increase the data size while training.

5.2.1 Setting S1

In setting S1, the trained model is expected to predict binding affinities for known drugs and targets. Therefore, the test set must consist of drugs and targets which were seen by the model during the training phase. This doesn't mean that the exact drug-target pairs must be present in both training and test sets but the individual drugs and targets. The training set is allowed to have other drugs and targets apart from the ones that are in the test set.

To ensure this, out of all the drug target pairs in the binding affinity dataset, 20% was randomly selected for the test dataset and the remaining 80% was taken as the initial training pool. From the pairs in the training pool, 10% was selected as the first sample ensuring that all the drugs and targets in the test set are present in that sample.

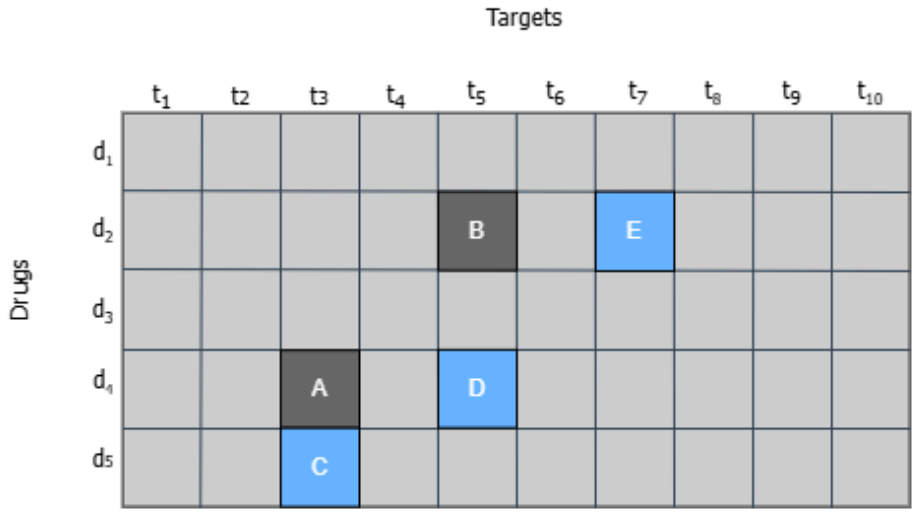

Figure 5.1: Selecting the first training sample of the binding affinity dataset in setting s1

Let A and B be two drug-target pairs in the test set of setting S1 (figure 5.1). A consist of drug d_4 and target t_3. The first sample (the 10%) of the training dataset should include this drug and target. For instance, the pair C and D could represent the drug d_4 and target t_3. B is a combination of drug d_2 and target t_5. To represent drug d_2 E can be added to the sample and D is already in the sample which also represent the target t_5.

After the first training sample was selected, more pairs were gradually added to the subsequent samples increasing the sample size by 10% of the initial training dataset size until it reaches 100%. Since the first training sample is included in all the other samples it guarantees that all the drugs and targets in the test set are present in all the training samples. The training samples of the binding affinity

matrix selected in the S1 training phase are sparse. Figure 5.2 shows a selected training sample highlighted in blue, where every drug and every target in the whole training dataset is represented at least once.

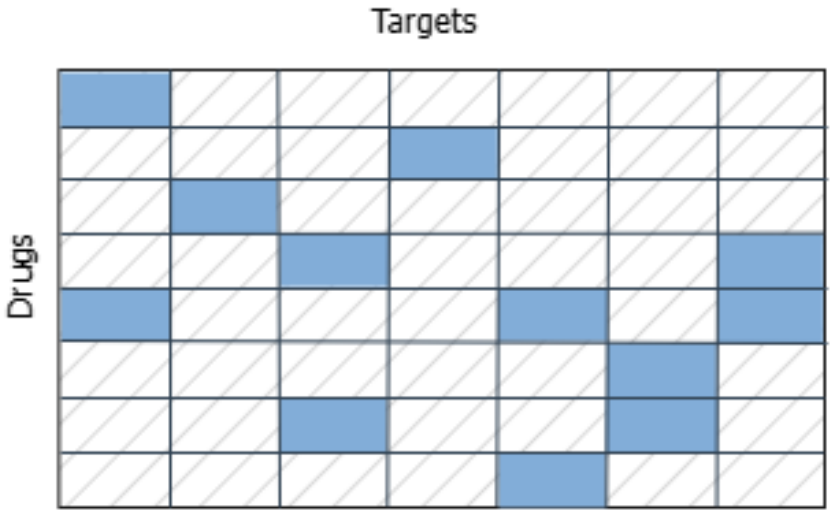

Figure 5.2: Illustration of sparse data in binding affinity training samples in setting S1

Complete drug-drug and target-target similarity matrices were provided in both training and testing phases.

5.2.2 Setting S2

In setting S2, the model is trained to predict binding affinities for novel drugs against known targets. Therefore, the complete target-target similarity matrix was used during both training and testing. As explained in section 5.1, we are treating the similarities as feature representations of drugs and targets. For that reason, the drug-drug similarity matrix was divided row wise into train and test with 80:20 ratio to include only the drugs that were present in the training sample of the drug target binding affinities (5.3).

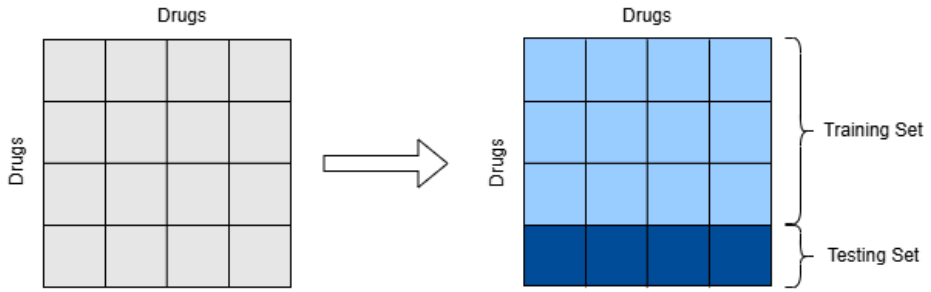

Figure 5.3: Train test split of drug-drug similarity matrix

The matrix which contains the drug target binding affinities were split by drugs only as the targets should be shared between the train and test sets (5.4).

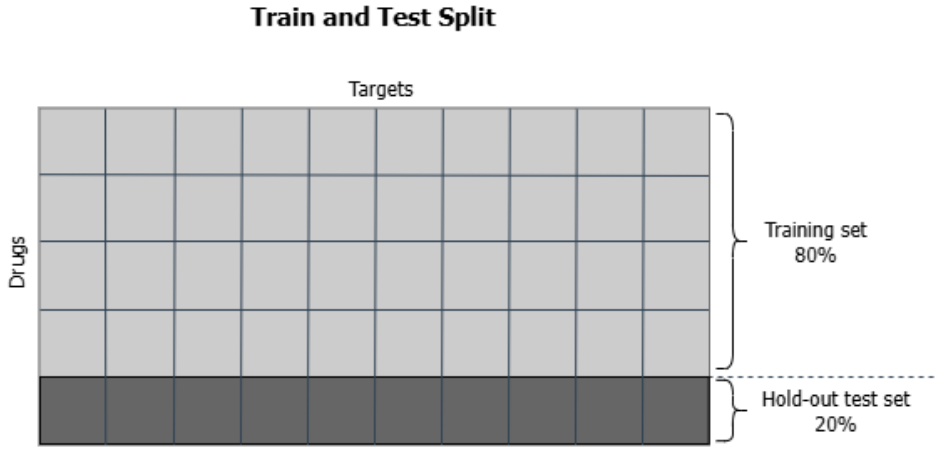

Figure 5.4: Train test split of binding affinity dataset in setting S2

During the training phase, both binding affinity and drug-drug similarity training datasets were again divided row-wise into 10 smaller samples (5.5).

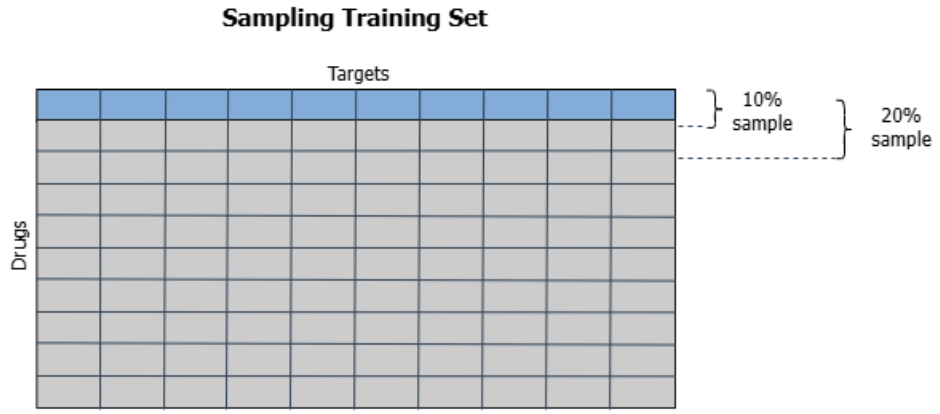

Figure 5.5: Sampling of the binding affinity training set in setting S2

5.2.3 Setting S3

In setting S3, the expectation is to train a model to predict binding affinities for novel targets against known drugs. To fulfill this requirement the complete drug-drug similarity matrix were used in both training and testing sets. However, the target-target similarity matrix was partitioned row-wise into training and testing to include only the targets that were present in the training sample of the drug target binding affinities (5.6).

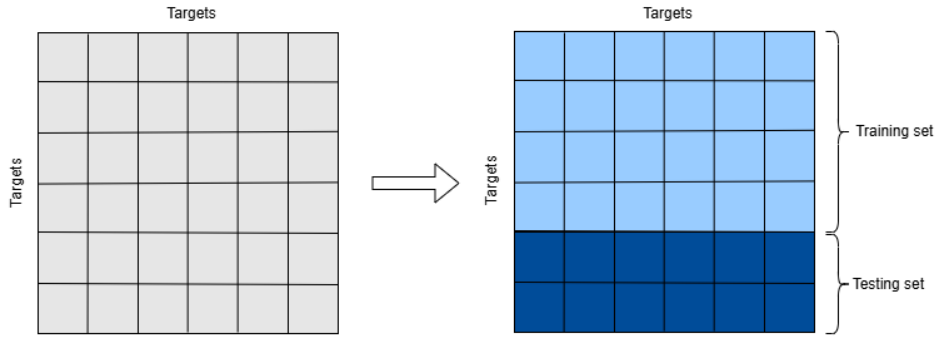

Figure 5.6: Train test split of target-target similarity matrix

The binding affinity dataset was split column-wise to ensure that the targets in the testing dataset are unseen to the model (5.7). Then, the training dataset of the binding affinities was again divided into smaller samples by targets. The target-

target dataset was sampled row-wise (5.8).

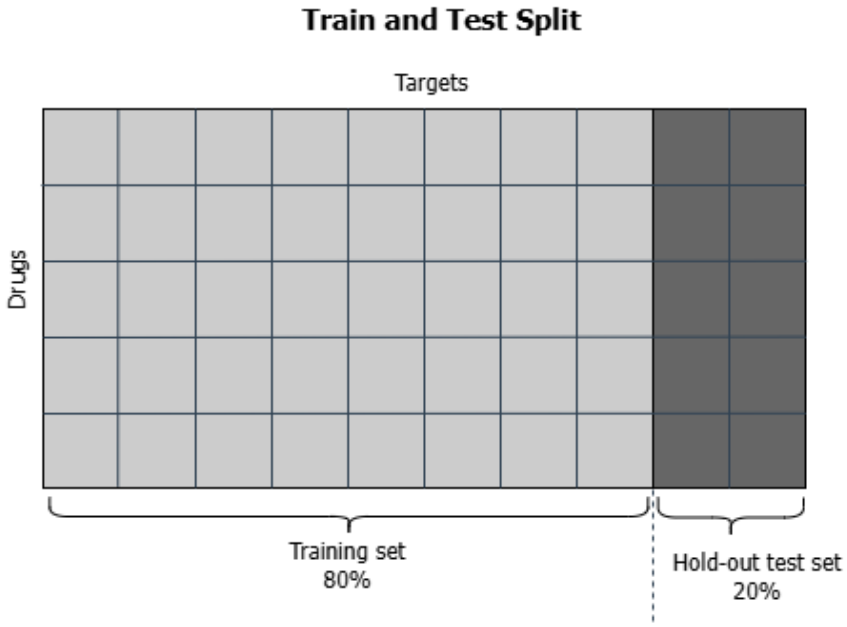

Figure 5.7: Train test split of binding affinity dataset in setting S3

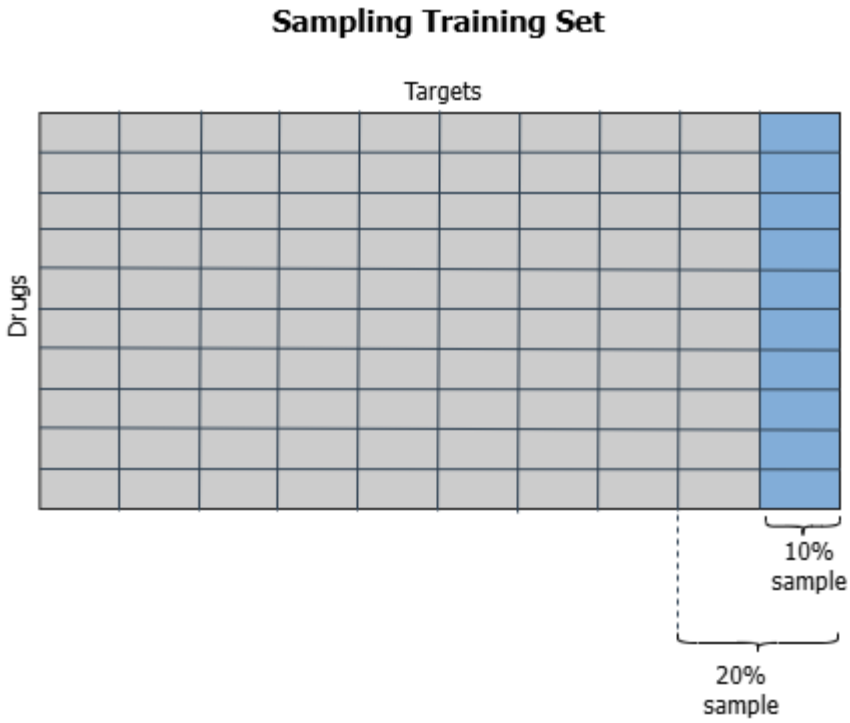

Figure 5.8: Sampling of the binding affinity training set in setting S3

5.2.4 Setting S4

In setting S4, the data sets were divided into train and test based on both drugs and targets such that there are no shared drugs or targets between the two datasets. Therefore, after the test set is separated, all the other drug-target pairs that involves either a drug or a target found in the test set must be removed from the remaining data in order to define the training dataset.

After removing the drug-target pairs as shown in 5.9, from the remaining data pairs the training and testing datasets were separated following 80:20 ratio. Let us define p_1 and p_2 as two fractions where,

$$p_1 + p_2 = 1 \tag{5.1}$$

When the number of drugs and targets in the dataset is given by n_d and n_t respec-

tively, as the training set is expected to be 80%,

$$\frac{n_d p_1 \times n_t p_1}{(n_d p_1 \times n_t p_1) + (n_d p_2 \times n_t p_2)} = \frac{80}{100}$$

After simplifying the above equation,

$$p_1 = 2p_2 \tag{5.2}$$

By solving the above two equations we get $p_1 \approx 0.67$ and $p_2 \approx 0.33$. Therefore, the training dataset was selected with 67% of both drugs and targets while the testing dataset has the remaining 33% of the drugs and targets.

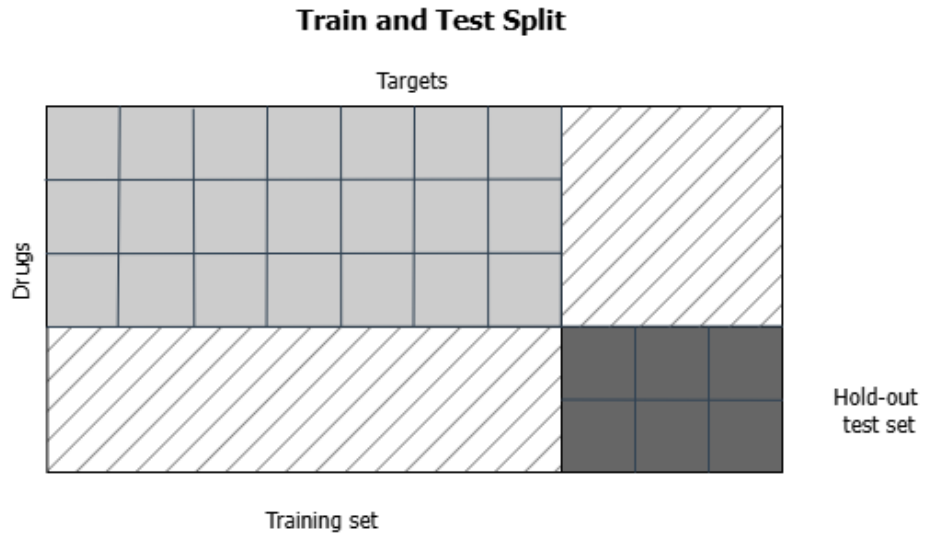

Figure 5.9: Train test split of setting S4

Figure 5.10: Sampling of the training set in setting S4

In this setting, both drugs and targets were incrementally added to the training data across iterations, rather than exposing the model to all 40 drugs or all the 300 targets in the data set starting from the first iteration (5.10).

It is important to understand how the dataset sizes vary in these four experimental settings as we are specifically trying to find out the impact of the dataset size on the models learning ability. In setting S4, part of the dataset pairs are removed to fulfill the setting conditions. Therefore it has the smallest training and testing datasets (5.1).

	Train		Test	
Experimental Setting	Drugs x Targets	Pairs	Drugs x Targets	Pairs
S1	68×442	24044	68×442	6011
S2	54×442	23868	14×442	6188
S3	68×354	24072	68×88	5984
S4	46×296	13616	22×146	3212

Table 5.1: Number of training and testing drug-target pairs under different experimental settings

5.3 Evaluation

Since the evaluation is performed using hold out test method, the results may be sensitive to the specific data partitioning. To avoid this, we repeat the same process with 100 different random splits for each setting.

The performance was measured by calculating the CI for each data sample across the 100 random splits. The mean CI and the standard deviation were then calculated for each training sample to capture both average performance and its variability.

The learning curves were plotted using the mean CI against the corresponding training dataset percentage with error bars representing the standard deviation. The learning curve for the training data was also included for comparison.

6 Results and Findings

This chapter presents the results and the analysis of the model performance across the different experimental settings. We performed four separate experiments training KronRLS linear pairwise predictors under different settings by systematically increasing the data size. In each iteration the trained model was evaluated using a hold out test set that was partitioned suitably for each setting, satisfying it's conditions.

6.1 Impact of Training Dataset Size on Overall Model Performance

In this section we investigate how the increasing training dataset size has influenced the overall performance of the model in each experimental setting focusing on the CI values reported in the starting and ending samples.

6.1.1 Setting S1

In setting S1, the model predicts binding affinities for known drugs and targets. It means that, during the training phase, the model has seen at least one interaction data involving each drug and each target in the test set.

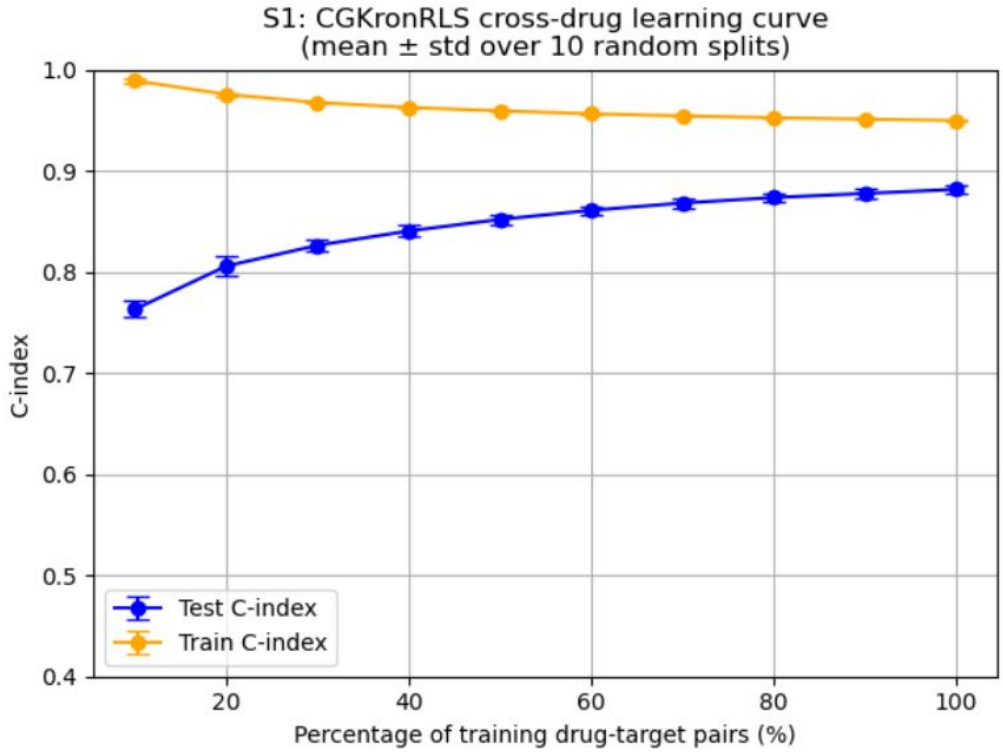

Figure 6.1: Training and Evaluation CI with increasing dataset size in setting S1. The model is showing the highest test CI 0.882 at 100% of the training data

As expected, setting S1 starts with a high CI value of 0.763 at 10% and shows significant improvement as the data size increases.

6.1.2 Setting S2

Shows a moderate improvement in test performance as the drug-target pairs in the training set increases.

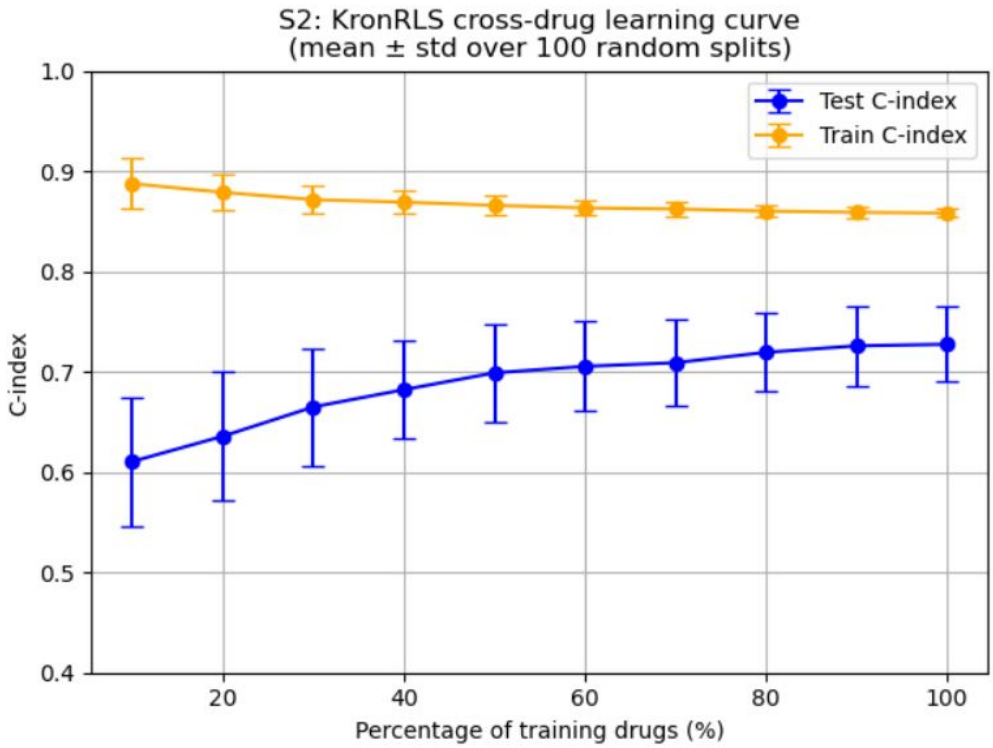

Figure 6.2: Training and Evaluation CI with increasing dataset size in setting S2. The model is showing the highest c-index 0.727 at 100% of the training data

The model trained under setting S2 shows only a small performance gain despite increasing data volume, with the CI rising only marginally from 0.611 to 0.727 as training pairs grow from 10% to 100%. It seems to struggle to do accurate predictions for new drugs in the test set.

6.1.3 Setting S3

The model exhibits a more consistent upward trend in CI, increasing from 0.693 to 0.819 showing better scalability with more training data.

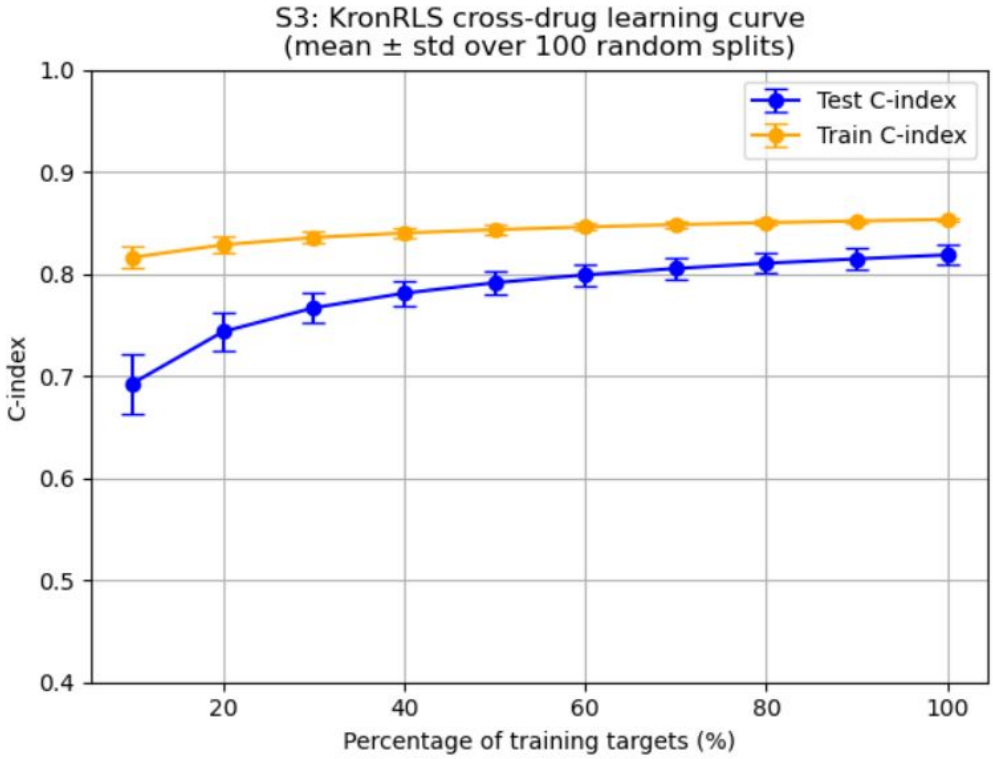

Figure 6.3: Training and Evaluation CI with increasing dataset size in setting S3. The model is showing the highest CI 0.819 at 100% of the training data

6.1.4 Setting S4

The model is showing a slight improvement in CI from 0.607 to 0.683. Shows a consistent improvement even though the minimum and maximum CI does not have a larger gap.

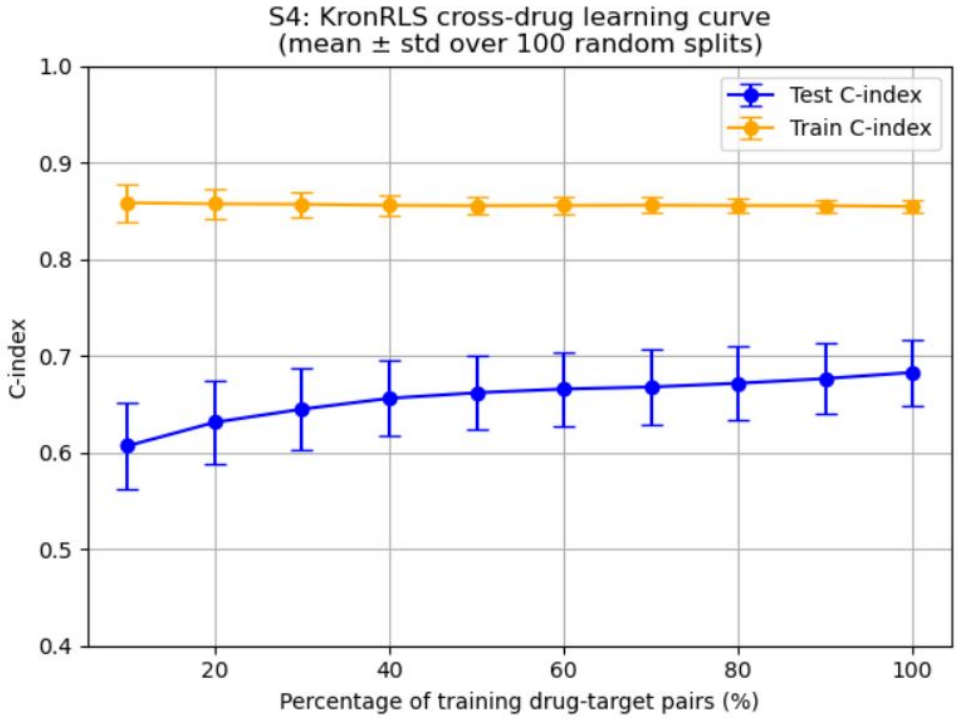

Figure 6.4: Training and Evaluation CI with increasing dataset size in setting S4. The model is showing the highest CI 0.683 at 100% of the training data

6.2 Analysis of The Shape of Learning Curves

In this section, we analyse the shape of the plotted learning curve of test data and see whether it reveals any trends.

- Setting S1: The learning curve shows the highest slope with the widest performance gap from start to end. Exhibits the most desirable learning behavior.
- Setting S2: The learning curve is erratic specially in the beginning of learning. Error bars are bigger.
- Setting S3: The test performance consistently improves with more data. The error bars are smaller which indicates that there is less variability in c-index values across the 10 different random splits. This can be identified as a well behaved learning curve, constantly showing an improvement in the performance.

- Setting S4:The learning curve looks smoother but it does not show a big slope.

Increasing number of training data pairs vary in the four experimental settings.

Experimental Setting	AULC	mean C-index at 10%	mean C-index at 100%
S1	0.7628	0.763	0.882
S2	0.6211	0.611	0.727
S3	0.7069	0.693	0.819
S4	0.5921	0.607	0.683

Table 6.1: Area Under the Learning Curve(AULC) values of each setting with the C-index values at starting and ending data samples

6.3 Evaluation of the Generalization Capability

In this section, we look at the shape of both training and testing curves to get insights on model's generalization capability.

- Setting S1: The model seems to generalize well to test data. Based on the shape of the curve, it appears that the model is approaching convergence.
- Setting S2: Shows a big gap between the training and testing curves suggesting overfitting and therefore weaker generalization.
- Setting S3: Demonstrates the narrowest gap implying strong generalization capability.
- Setting S4: Shows a wider gap indicating weaker generalization capability.

6.4 Comparison of Prediction Performance Among The Experimental Settings

The four experimental settings demonstrate different levels of prediction performance. As shown in table 5.1, each setting has different number of drug-target pairs

and the percentage in each setting doesn't mean the same number of drug-target pairs across the four settings. Therefore, the results are not directly comparable. The figure 6.5 is only used to get the general overall idea about the learning.

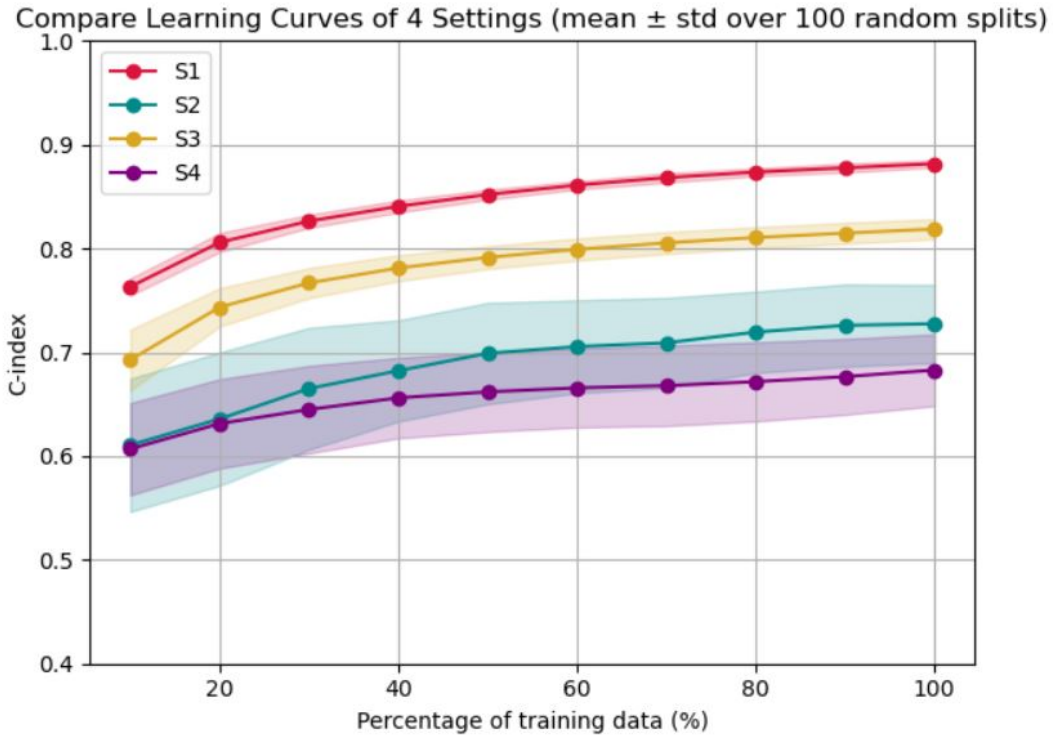

Figure 6.5: A comparison of the four learning curves

- Setting S1 model outperformed other settings achieving the highest CI both at the start and end of the training. This shows that the model has benefited from knowing some interaction information on drugs and targets.

- Secondly, S3 outperforms both S2 and S4 settings in terms of highest c-index. Although S3 represents a challenging setting than S1. the model showed consistency in model's learning capability. The shape of the S3 learning curve is very similar to the learning curve observed in S4 except with lower values. This suggests that the model is relatively good at leveraging knowledge from known interactions to predict binding affinities for unseen targets.

- Setting S2 was able to achieve slightly better peak performance than S4, but this setting displays a slightly unstable learning curve compared to others. Among all settings, the learning curve of S2 appears to be the most sensitive setting to the data split particularly in the first few samples of the experiment. This suggests that the model performance was not much stable.

- Setting S4 reports the lowest CI value out of the four experimental settings. Initially, both S2 and S4 exhibits roughly similar values. However, as the training progresses, the performance of the S4 only shows a smaller improvement. When compared with S2, the learning appears to be relatively stable in S4. The curve indicates a very slow learning which shows that the model is having difficulty to capture enough patterns to predict interactions for novel drugs and targets.

7 Discussion

This study aimed to investigate the influence of training dataset size on performance of drug-target interaction binding affinity prediction model under four experimental settings (S1-S4). As the model, KronRLS linear pairwise predictor was used. The results of the experiment revealed that the performance of the model varies significantly across settings. It also gave insights on the impact of the dataset size on models generalization capability.

All the settings appears to improve their performance while some experimental settings show a significant improvement. Setting S1 and S3 seems that the model is closer to approaching convergence with the available data. Setting S2 and S4 are only showing slight overall improvement. However, as the curve is showing an upward trend, we can assume that more data might benefit the model to improve its performance in these two settings.

Setting S1, which is the most easiest prediction task for the model out of the four settings, showed the highest performance in this experiment. In this setting the model gets to know at least one interaction about every drug and target it is predicting interactions in the testing phase.

Setting S2 and S3 are scenarios where at least one of the prediction entities (drug or target) is already known to the model from the training phase. The model in S2 which is trained to predict binding affinities for unknown drugs, showed some difficulty to keep the performance constantly increasing. The learning curve dis-

played fluctuations with relatively large error bars reflecting the affect of random data splits. The Davis data consist of 68 drugs and 442 targets. This imbalance can be a reason for the model to face difficulty when trying to predict for new drugs. Also, in this study, the model relies on drug-drug similarities derived from Tanimoto coefficient which is based on the chemical structure of a drug. The structural similarity of two drugs may not always indicate functional similarity. There are drugs which are chemically similar without similar pharmacological properties as well as chemically different but with similar pharmacological behaviors. [44]. This may also be a reason behind the difficulty as the predictions solely rely on structural similarity of drugs.

The model of setting S3, which is trained to predict binding affinity values for unknown targets was showing a smooth increasing learning curve throughout with narrow error bars. S3, also reported the highest c-index value among the settings which involved unseen drugs and targets. Out of all the experimental settings, setting S3 has the highest number of drug target pairs in it's initial training dataset (refer to 5.1). Interestingly, although the target-target similarities are generally low across the data distribution, it doesn't appear to negatively affect the model performance in this setting. When compared with setting S2, S3 is better at generalizing to new targets and similar observation was reported by Pahikkala et al. [5].

The most challenging scenario out of the four settings is S4, where the model was trained to predict binding affinities for unknown drugs and unknown targets. This reported the lowest c-index value out of all settings. It is also important to note that this setting has the smallest number of drug-target pairs as a subset of pairs was removed from the dataset to ensure neither targets nor drugs in test set appeared in the training data. However, a slight improvement can be seen with the increase of the data size. The gap between the training and testing curves highlights the limited ability to generalize to novel drugs and targets with no interaction data.

Overall, these observations emphasize the importance of evaluating prediction models under these realistic experimental settings as the performance can be over-estimated. These results also highlights the ability of KronRLS algorithm to perform well in binding affinity prediction of the setting S3 which involves unknown targets.

8 Conclusion

This thesis explored the impact of training dataset size on the performance of a KronRLS model in drug-target interaction binding affinity prediction. The study focused on four experimental settings(S1-S4) that represent realistic scenarios that reflect the actual challenges faced in the field of DTI prediction. A major challenge in DTI studies is the limited availability of known interactions among drugs and targets. Similarity based approaches, such as KronRLS, rely on leveraging these known interactions to predict unknown interactions with the use of chemical space and genomic spaces. Treating the DTIs as a regression task, it assess the interaction strengths, providing more informative insights than binary classification tasks. The model performance was evaluated by analysing the shape of the learning curves.

The results of this experiment demonstrated that the model performance is highly dependent on the specific experimental setting depending on whether the test data contains known or unseen drugs and targets. By analysing learning curves, the study revealed how the KronRLS model responds differently to increasing training data size under each setting emphasizing its strengths and limitations. As expected, setting S1 reported the highest accuracy and demonstrated good learning abilities achieving relatively high performance starting from the smallest training sample. Notably, setting S3 which predicts interactions for new targets reported the second highest performance and generalization ability. In contrast, Setting S2 which involves predicting interactions for new drugs, indicated a problem of overfitting

and difficulty maintaining consistent learning. Setting S4 which is the most challenging setting involving both unseen drugs and targets also wasn't able to improve the performance much with increasing training data size and showed difficulty with generalizing to unseen drugs and targets. This thesis highlights the importance of distinguishing among the four experimental settings in DTI prediction and understanding their data size requirements.

While this provides valuable insights, it is not without limitations. This experiment were conducted using a single dataset and a single ML technique. As this study is using linear kernels the model is limited to capturing linear relationships in data. By expanding this experiment to include multiple datasets with different kernels could provide deeper insights and broaden the scope of the findings.

9 Statement on the Use of AI in the Thesis

In the writing of this thesis, I have used few AI tools primarily to improve the language. Specifically I used ChatGPT (https://openai.com/chatgpt/overview/) and DeepSeek (https://deep-seek.chat/) to rephrase sentences into a more academic style and to get suggestions on how to structure the content of the thesis meaningfully. I have also used QuillBot (https://quillbot.com/) and ChatGPT to get ideas on how to paraphrase sentences while doing the literature review. However, I did not use any information generated by these AI tools directly and without reviewing.

References

[1] T. T. Ashburn and K. B. Thor, "Drug repositioning: Identifying and developing new uses for existing drugs", *Nature reviews Drug discovery*, vol. 3, no. 8, pp. 673–683, 2004.

[2] E. H. B. Maia, L. C. Assis, T. A. De Oliveira, A. M. Da Silva, and A. G. Taranto, "Structure-based virtual screening: From classical to artificial intelligence", *Frontiers in chemistry*, vol. 8, p. 343, 2020.

[3] G. Pujadas et al., "Protein-ligand docking: A review of recent advances and future perspectives", *Current Pharmaceutical Analysis*, vol. 4, no. 1, pp. 1–19, 2008.

[4] P. Ripphausen, B. Nisius, and J. Bajorath, "State-of-the-art in ligand-based virtual screening", *Drug discovery today*, vol. 16, no. 9-10, pp. 372–376, 2011.

[5] T. Pahikkala et al., "Toward more realistic drug–target interaction predictions", *Briefings in bioinformatics*, vol. 16, no. 2, pp. 325–337, 2015.

[6] J.-L. Reymond, R. Van Deursen, L. C. Blum, and L. Ruddigkeit, "Chemical space as a source for new drugs", *MedChemComm*, vol. 1, no. 1, pp. 30–38, 2010.

[7] J. L. Medina-Franco, K. Martínez-Mayorga, M. A. Giulianotti, and R. A. Houghten, "Visualization of the chemical space in drug discovery", *Current Computer-Aided Drug Design*, vol. 4, no. 4, pp. 322–333, 2008.

[8] Y. Yamanishi, "Chemogenomic approaches to infer drug–target interaction networks", *Data Mining for Systems Biology: Methods and Protocols*, pp. 97–113, 2013.

[9] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2022.

[10] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning. arxiv 2018", *arXiv preprint arXiv:1811.12808*, 2021.

[11] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science", *Patterns*, vol. 4, no. 9, 2023.

[12] F. Mohr and J. N. van Rijn, "Learning curves for decision making in supervised machine learning: A survey", *Machine Learning*, vol. 113, no. 11, pp. 8371–8425, 2024.

[13] T. Viering and M. Loog, "The shape of learning curves: A review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7799–7819, 2022.

[14] C. Perlich, F. Provost, and J. S. Simonoff, "Tree induction vs. logistic regression: A learning-curve analysis", *Journal of Machine Learning Research*, vol. 4, no. Jun, pp. 211–255, 2003.

[15] M. Viljanen, A. Airola, and T. Pahikkala, "Generalized vec trick for fast learning of pairwise kernel models", *Machine Learning*, vol. 111, no. 2, pp. 543–573, 2022.

[16] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning", 2008.

[17] T. Pahikkala and A. Airola, "Rlscore: Regularized least-squares learners", *Journal of Machine Learning Research*, vol. 17, no. 220, pp. 1–5, 2016. [Online]. Available: `http://jmlr.org/papers/v17/16-470.html`.

[18] A. Danchin, "In vivo, in vitro and in silico: An open space for the development of microbe-based applications of synthetic biology", *Microbial Biotechnology*, vol. 15, no. 1, pp. 42–64, 2022.

[19] K. Ullrich, J. Mack, and P. Welke, "Ligand affinity prediction with multi-pattern kernels", in *International Conference on Discovery Science*, Springer, 2016, pp. 474–489.

[20] H. Yu et al., "A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data", *PloS one*, vol. 7, no. 5, e37608, 2012.

[21] M. L. MacDonald et al., "Identifying off-target effects and hidden phenotypes of drugs in human cells", *Nature chemical biology*, vol. 2, no. 6, pp. 329–337, 2006.

[22] E. Lounkine et al., "Large-scale prediction and testing of drug activity on side-effect targets", *Nature*, vol. 486, no. 7403, pp. 361–367, 2012.

[23] Y. Yamanishi, E. Pauwels, and M. Kotera, "Drug side-effect prediction based on the integration of chemical and biological spaces", *Journal of chemical information and modeling*, vol. 52, no. 12, pp. 3284–3292, 2012.

[24] H. Ding, I. Takigawa, H. Mamitsuka, and S. Zhu, "Similarity-based machine learning methods for predicting drug–target interactions: A brief review", *Briefings in bioinformatics*, vol. 15, no. 5, pp. 734–747, 2014.

[25] R. Chen, X. Liu, S. Jin, J. Lin, and J. Liu, "Machine learning for drug-target interaction prediction", *Molecules*, vol. 23, no. 9, p. 2208, 2018.

[26] N. Nagamine and Y. Sakakibara, "Statistical prediction of protein–chemical interactions based on chemical structure and mass spectrometry data", *Bioinformatics*, vol. 23, no. 15, pp. 2004–2012, 2007.

[27] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces", *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, 2008.

[28] M. Bagherian, E. Sabeti, K. Wang, M. A. Sartor, Z. Nikolovska-Coleska, and K. Najarian, "Machine learning approaches and databases for prediction of drug–target interaction: A survey paper", *Briefings in bioinformatics*, vol. 22, no. 1, pp. 247–269, 2021.

[29] M. C. Cobanoglu, C. Liu, F. Hu, Z. N. Oltvai, and I. Bahar, "Predicting drug–target interactions using probabilistic matrix factorization", *Journal of chemical information and modeling*, vol. 53, no. 12, pp. 3399–3409, 2013.

[30] F. Cheng et al., "Prediction of drug-target interactions and drug repositioning via network-based inference", *PLoS computational biology*, vol. 8, no. 5, e1002503, 2012.

[31] T. Van Laarhoven, S. B. Nabuurs, and E. Marchiori, "Gaussian interaction profile kernels for predicting drug–target interaction", *Bioinformatics*, vol. 27, no. 21, pp. 3036–3043, 2011.

[32] M. I. Davis et al., "Comprehensive analysis of kinase inhibitor selectivity", *Nature biotechnology*, vol. 29, no. 11, pp. 1046–1051, 2011.

[33] H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: Deep drug–target binding affinity prediction", *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.

[34] H. Öztürk, E. Ozkirimli, and A. Özgür, "Widedta: Prediction of drug-target binding affinity", *arXiv preprint arXiv:1902.04166*, 2019.

[35] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "Graphdta: Predicting drug–target binding affinity with graph neural networks", *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 2021.

[36] M. Stock, T. Pahikkala, A. Airola, B. De Baets, and W. Waegeman, "A comparative study of pairwise learning methods based on kernel ridge regression", *Neural Computation*, vol. 30, pp. 2245–2283, 2018.

[37] A. M. Wassermann, H. Geppert, and J. Bajorath, "Ligand prediction for orphan targets using support vector machines and various target-ligand kernels is dominated by nearest neighbor effects", *Journal of chemical information and modeling*, vol. 49, no. 10, pp. 2155–2167, 2009.

[38] L. Jacob and J.-P. Vert, "Protein-ligand interaction prediction: An improved chemogenomics approach", *bioinformatics*, vol. 24, no. 19, pp. 2149–2156, 2008.

[39] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting protein–protein interactions", *Bioinformatics*, vol. 21, no. suppl_1, pp. i38–i46, 2005.

[40] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester, "Simboost: A read-across approach for predicting drug–target binding affinities using gradient boosting machines", *Journal of cheminformatics*, vol. 9, pp. 1–14, 2017.

[41] S. Vilar et al., "Similarity-based modeling in large-scale prediction of drug-drug interactions", *Nature protocols*, vol. 9, no. 9, pp. 2147–2163, 2014.

[42] J. Menke, J. Massa, and O. Koch, "Natural product scores and fingerprints extracted from artificial neural networks", *Computational and Structural Biotechnology Journal*, vol. 19, pp. 4593–4602, 2021.

[43] T. F. Smith, M. S. Waterman, et al., "Identification of common molecular subsequences", *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[44] C. G. Wermuth, "Similarity in drugs: Reflections on analogue design", *Drug Discovery Today*, vol. 11, no. 7-8, pp. 348–354, 2006.