

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/322851488>

# DeepDTA: Deep Drug-Target Binding Affinity Prediction

**Article** · January 2018

CITATIONS

0

READS

115

## 3 authors:



**Hakime Öztürk**

Bogazici University

8 PUBLICATIONS 27 CITATIONS

[SEE PROFILE](#)



**Elif Ozkirimli**

Bogazici University

46 PUBLICATIONS 248 CITATIONS

[SEE PROFILE](#)



**Arzucan Ozgur**

Bogazici University

78 PUBLICATIONS 1,007 CITATIONS

[SEE PROFILE](#)

## Some of the authors of this publication are also working on these related projects:



Development of an integrated methodology for drug discovery by virtual screening and docking [View project](#)



Ligand centric approach to drug discovery [View project](#)

# DeepDTA: Deep Drug-Target Binding Affinity Prediction

Hakime Öztürk<sup>1,\*</sup>, Elif Ozkirimli<sup>2,\*</sup>, and Arzucan Özgür<sup>1,\*</sup>

**1 Department of Computer Engineering, Bogazici University, Istanbul, 34342, Turkey**

**2 Department of Chemical Engineering, Bogazici University, Istanbul, 34342, Turkey**

\* [hakime.ozturk@boun.edu.tr](mailto:hakime.ozturk@boun.edu.tr); [elif.ozkirimli@boun.edu.tr](mailto:elif.ozkirimli@boun.edu.tr);  
[arzucan.ozgur@boun.edu.tr](mailto:arzucan.ozgur@boun.edu.tr)

## Abstract

The identification of novel drug-target (DT) interactions is a substantial part of the drug discovery process. Most of the computational methods that have been proposed to predict DT interactions have focused on binary classification, where the goal is to determine whether a DT pair interacts or not. However, protein-ligand interactions assume a continuum of binding strength values, also called **binding affinity** and predicting this value still remains a challenge. The increase in the affinity data available in DT knowledge-bases allows the use of advanced learning techniques such as deep learning architectures in the prediction of binding affinities. In this study, we propose a **deep-learning based model that uses only sequence information of both targets and drugs to predict DT interaction binding affinities**. The few studies that focus on DT binding affinity prediction use either 3D structures of protein-ligand complexes or 2D features of compounds. One novel approach used in this work is the modeling of protein sequences and compound 1D representations with convolutional neural networks (CNNs). The results show that the proposed deep learning based model that uses the 1D representations of targets and drugs is an effective approach for drug target binding affinity prediction. The model in which high-level representations of a drug and a target are constructed via CNNs achieved the best Concordance Index (CI) performance in one of our larger benchmark data sets, outperforming the KronRLS algorithm and SimBoost, a state-of-the-art method for DT binding affinity prediction.

## Introduction

The successful identification of drug-target interactions (DTI) is a critical step in drug discovery. As the field of drug discovery expands with the discovery of new drugs, repurposing of existing drugs and identification of novel interacting partners for approved drugs is also gaining interest [44]. Until recently, DTI prediction was approached as a binary classification problem [4, 6, 7, 15, 25, 45, 56, 61], neglecting an important piece of information about protein-ligand interactions, namely the binding affinity values. Binding affinity provides information

on the strength of the interaction between a drug-target (DT) pair and it is usually expressed in measures such as dissociation constant ( $K_d$ ), inhibition constant ( $K_i$ ), or the half maximal inhibitory concentration (IC50). IC50 depends on the concentration of the target and ligand [8] and low IC50 values signal strong binding. Similarly, low  $K_i$  values indicate high binding affinity.  $K_d$  and  $K_i$  values are usually represented in terms of  $pK_d$  or  $pK_i$ , the negative logarithm of the dissociation or inhibition constants.

In binary classification based DTI prediction studies, construction of the data sets constitutes a major problem, since negative (not-binding) information is generally hard to find. In most cases, the DT pairs for which binding information is not known are treated as negative (not-binding) samples. The lack of true-negative samples and how the study generates synthetic negative samples usually affects the performance of the prediction algorithms. On the other hand, formulating the DT prediction task as a binding affinity prediction problem enables the creation of more realistic data sets, where the binding affinity scores are directly used, obviating the need for the generation of synthetic negative samples.

Prediction of protein-ligand binding affinities has been the focus of protein-ligand scoring, which is frequently used after virtual screening and docking campaigns in order to predict the putative strengths of the proposed ligands to the target [48]. Non-parametric machine learning methods such as the Random Forest (RF) algorithm have been used as a successful alternative to scoring functions that depend on multiple parameters [3, 38, 50]. However, Gabel et al. showed that RF-score failed in virtual screening and docking tests, speculating that using features such as co-occurrence of atom-pairs over-simplified the description of the protein-ligand complex and led to the loss of information that the raw interaction complex could provide [22]. Around the same time this study was published, deep learning started to become a popular architecture powered by the increase in data and high capacity computing machines challenging machine learning methods.

Inspired by the remarkable success rate in image processing [14, 18, 51] and speech recognition [16, 27, 30], deep learning methods are now being intensively used in many other research fields, including bioinformatics such as in genomics studies [37, 60] and quantitative-structure activity relationship (QSAR) studies in drug discovery [40]. The major advantage of deep learning architectures is that they enable better representations of the raw data by non-linear transformations in each layer [36] and thus they facilitate learning the hidden patterns in the data.

A few studies employing Deep Neural Networks (DNN) have already been performed for DTI binary class prediction using different input models for proteins and drugs [9, 28, 55] in addition to some studies that employ stacked auto-encoders [58] and deep-belief networks [59]. Similarly, stacked auto-encoder based models with Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) were applied to represent chemical and genomic structures in real-valued vector forms [24, 32]. Deep learning approaches have also been applied to protein-ligand interaction scoring in which a common application has been the use of CNNs that learn from the 3D structures of the protein-ligand complexes [23, 48, 57]. However, this approach is limited to known protein-ligand complex structures, with only 25000 ligands reported in PDB [49].

Pahikkala et al. employed the Kronecker Regularized Least Squares (KronRLS) algorithm that utilizes only 2D based compound similarity-based representations of the drugs and Smith-Waterman similarity representation of the targets [47]. Recently, SimBoost method was proposed to predict binding affinity scores with a gradient boosting machine by using

feature engineering to represent drug-target interactions [29]. They utilized similarity-based information of DT pairs as well as features that were extracted from network-based interactions between the pairs. Both studies used traditional machine learning algorithms and utilized 2D-representations of the compounds in order to obtain similarity information.

In this study, we propose an approach to predict the binding affinities of protein-ligand interactions with deep learning models using only sequences (1D representations) of proteins and ligands. **To this end, the sequences of the proteins and SMILES (Simplified Molecular Input Line Entry System) representations of the compounds are used rather than external features or 3D-structures of the binding complexes.** We employ CNN blocks to learn representations from the raw protein sequences and SMILES strings and combine these representations to feed into a fully-connected layer block that we call DeepDTA. We use the Davis Kinase binding affinity data set [17] and the KIBA large-scale kinase inhibitors bioactivity data [29, 54] to evaluate the performance of our model and compare our results with the KronRLS [47] and SimBoost algorithms [29]. Our new model that uses two separate CNN-based blocks to represent proteins and drugs performs as well as the KronRLS and SimBoost algorithms on the Davis data set, and it performs significantly better than both the KronRLS and SimBoost algorithms on the KIBA data (p-value, 0.0001). With our proposed model, we also obtain the lowest Mean Squared Error (MSE) value on both data sets.

## Materials and Methods

### Data sets

We evaluated our proposed model on two different data sets, the Kinase data set Davis [17] and KIBA data set [54], which were previously used as benchmark data sets for binding affinity prediction evaluation [29, 47].

The Davis data set contains selectivity assays of the kinase protein family and the relevant inhibitors with their respective dissociation constant ( $K_d$ ) values. It comprises interactions of 442 proteins and 68 ligands. The KIBA data set, on the other hand, originated from an approach called KIBA, in which kinase inhibitor bioactivities from different sources such as  $K_i$ ,  $K_d$ , and  $IC_{50}$  were combined [54]. KIBA scores were constructed to optimize the consistency between  $K_i$ ,  $K_d$ , and  $IC_{50}$  by utilizing the statistical information they contained. The KIBA data set originally comprised 467 targets and 52498 drugs. [29] filtered it to contain only drugs and targets with at least 10 interactions yielding a total of 229 unique proteins and 2111 unique drugs. Table 1 summarizes these data sets in the forms that we used in our experiments.

**Table 1.** Data set

	Proteins	Compounds	Interactions
Davis ( $K_d$ )	442	68	30056
KIBA	229	2111	118254

While [47] used the  $K_d$  values of the Davis data set directly as the binding affinity values, we used the values transformed into log space,  $pK_d$ , similar to [29] as explained in Equation

1.

$$pK_d = -\log_{10}\left(\frac{K_d}{1e9}\right) \quad (1)$$

Figure 1A (left panel) illustrates the distribution of the binding affinity values in  $pK_d$  form. The peak at  $pK_d$  value 5 ( $10000nM$ ) constitutes more than half of the data set (20931 out of 30056). These values correspond to the negative pairs that either have very weak binding affinities ( $K_d > 10000nM$ ) or are not observed in the primary screen [47]. As such they are true negatives.

The distribution of the KIBA scores is depicted in the right panel of Figure 1A. [29] pre-processed the KIBA scores as follows: (i) for each KIBA score, its negative was taken, (ii) the minimum value among the negatives was chosen, and (iii) the absolute value of the minimum was added to all negative scores, thus constructing the final form of the KIBA scores.

The compound SMILES strings of the Davis data set were extracted from the Pubchem compound database based on their Pubchem CIDs [5]. For KIBA, first the ChEMBL IDs were converted into Pubchem CIDs and then, the corresponding CIDs were used to extract the SMILES strings. Figure 1B illustrates the distribution of the lengths of the SMILES strings of the compounds in the Davis (left) and KIBA (right) data sets. For the compounds of the Davis dataset, the maximum length of a SMILES is 103, while the average length is equal to 64. For the compounds of KIBA, the maximum length of a SMILES is 590, while the average length is equal to 58.

The protein sequences of the Davis data set were extracted from the UniProt protein database based on gene names/RefSeq accession numbers [2]. Similarly, the UniProt IDs of the targets in the KIBA data set were used to collect the protein sequences. Figure 1C left panel shows the lengths of the sequences of the proteins in the Davis data set. The maximum length of a protein sequence is 2549 and the average length is 788 characters. Figure 1C right panel depicts the distribution of protein sequence length in KIBA targets. The maximum length of a protein sequence is 4128 and the average length is 728 characters.

We should also note that the Smith-Waterman (S-W) similarity among proteins of the KIBA data set is at most 60% for 99% of the protein pairs. The target similarity is at most 60% for 92% of the protein pairs for the Davis data set. These statistics indicate that both data sets are non-redundant.

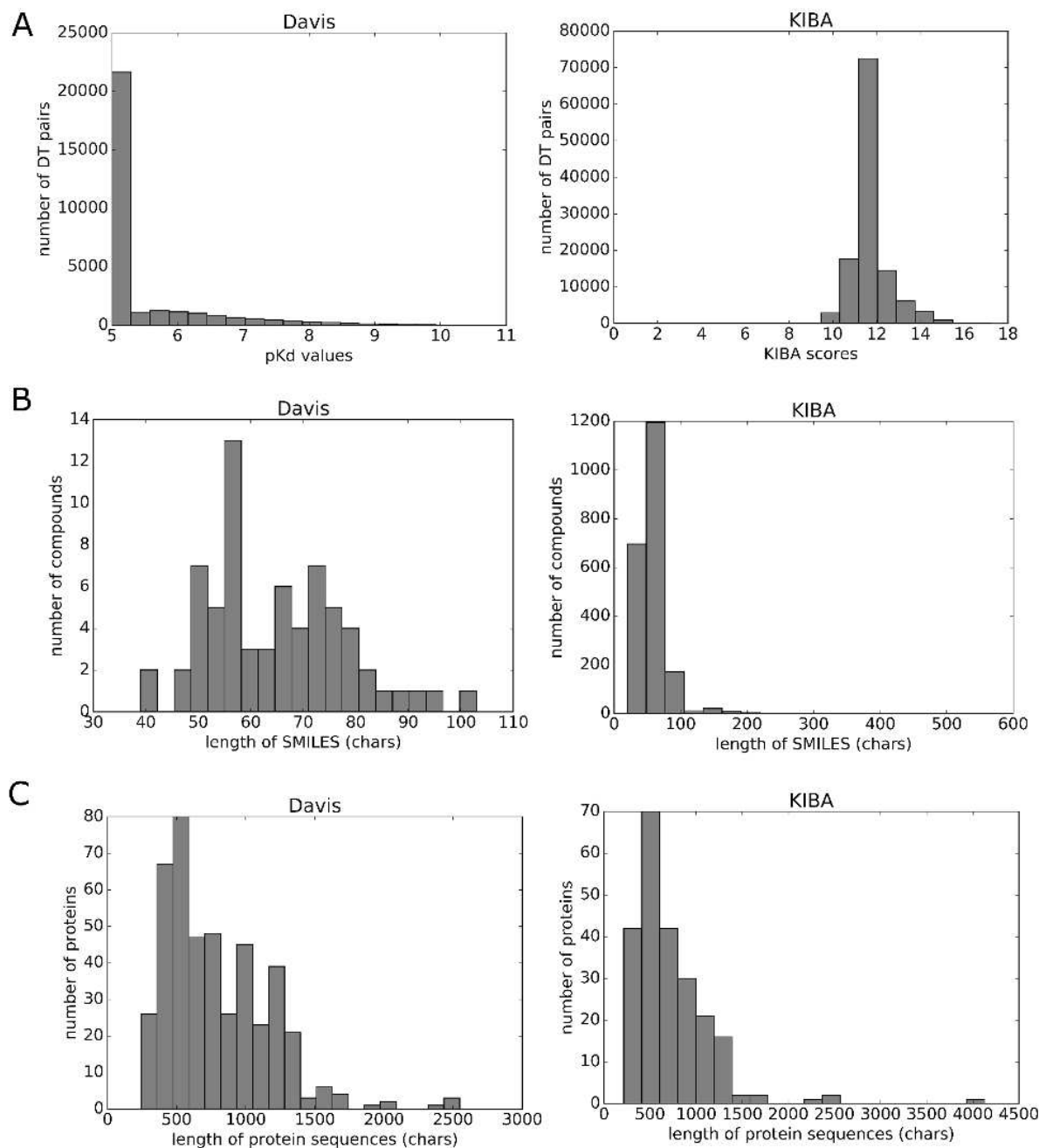
## Input Representation

We used integer/label encoding that uses integers for the categories to represent inputs. We scanned approximately 2M SMILES sequences that we collected from Pubchem and compiled 64 labels (unique letters). For protein sequences, we scanned 550K protein sequences from UniProt and extracted 25 categories (unique letters).

Here we represent each label with a corresponding integer (e.g. "C":1, "H":2, 'N':3 etc.). The label encoding for the example SMILES, "CN=C=O", is given below.

$$[C \ N \ = \ C \ = \ O] = [1 \ 3 \ 63 \ 1 \ 63 \ 5]$$

Protein sequences are encoded in a similar way using label encodings. Both SMILES and protein sequences have varying lengths. Hence, in order to create an effective representation

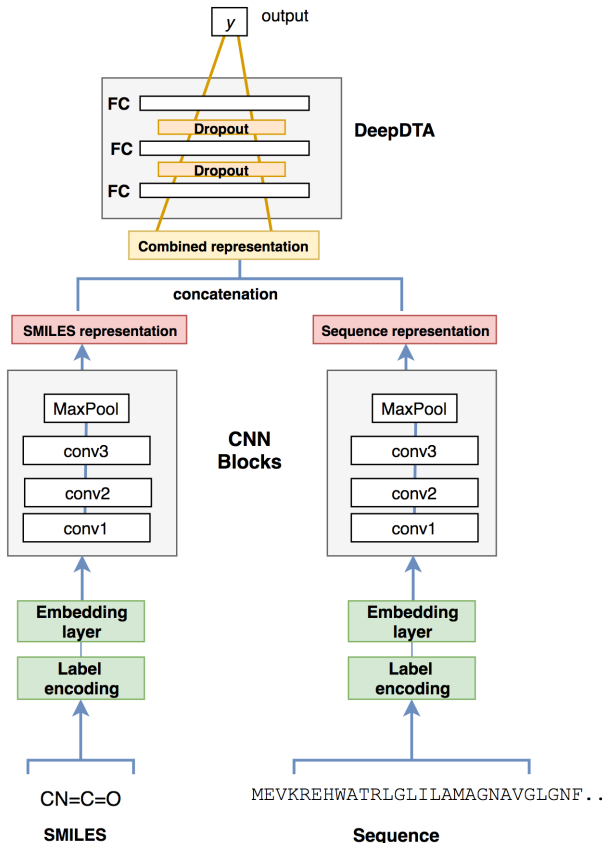


**Figure 1.** Summary of the Davis (left panel) and KIBA (right panel) data sets. A) Distribution of binding affinity values B) Distribution of the lengths of the SMILES strings C) Distribution of the lengths of the protein sequences .

form, we decided on fixed maximum lengths of 85 for SMILES and 1200 for protein sequences for Davis. To represent the components of KIBA, we chose the maximum 100 characters length for SMILES and 1000 for protein sequences. We chose these maximum lengths based on the distributions illustrated in Figure 1B and 1C so that the maximum lengths cover at least 80% of the proteins and 90% of the compounds in the data sets. The sequences that are longer than the maximum length are truncated, whereas shorter sequences are 0-padded.

## Proposed Model

In this study we treated protein-ligand interaction prediction as a regression problem by aiming to predict the binding affinity scores. As a prediction model, we adopted a popular deep learning architecture, Convolutional Neural Network (CNN). CNN is an architecture that contains one or more convolutional layers often followed by a pooling layer. A pooling layer down-samples the output of the previous layer and provides a way of generalization of the features that are learned by the filters. On top of the convolutional and pooling layers, the model is completed with one or more fully connected (FC) layers. The most powerful feature of CNN models is their ability to capture the local dependencies with the help of filters. Therefore, the number and size of the filters in a CNN directly affects the type of features the model learns from the input. It is often reported that as the number of filters increases, the model becomes better at recognizing patterns [33].



**Figure 2.** DeepDTA model with two CNN blocks to learn from compound SMILES and protein sequences.

We proposed a CNN-based prediction model that comprises two separate CNN blocks, each of which aims to learn representations from SMILES strings and protein sequences. For each CNN block, we used three consecutive 1D-convolutional layers with increasing number of filters. The second layer had double and the third convolutional layer had triple the number of filters in the first one. The convolutional layers were then followed by the max-pooling layer. The final features of the max-pooling layers were concatenated and fed into three FC layers, which we named as DeepDTA. We used 1024 nodes in the first two FC layers, each followed by a dropout layer of rate 0.1. Dropout is a regularization technique that is used to avoid over-fitting by setting the activation of some of the neurons to 0 [53]. The third layer consisted of 512 nodes and was followed by the output layer. The proposed model that combines two CNN blocks is illustrated in Figure 2.

As the activation function, we used **Rectified Linear Unit (ReLU)** [42],  $g(x) = \max(0, x)$ , which has been widely used in deep learning studies [36]. A learning model tries to minimize the difference between the expected (real) value and the prediction during training. Since we work on a regression task, we used mean squared error (MSE) as the loss function, in which  $P$  is the prediction vector, and  $Y$  corresponds to the vector of actual outputs.  $n$  indicates the number of samples.

$$MSE = \frac{1}{n} \sum_{i=1}^n (P_i - Y_i)^2 \quad (2)$$

The learning was completed with 100 epochs and mini-batch size of 256 was used to update the weights of the network. Adam was used as the optimization algorithm to train the networks [35] with the default learning rate of 0.001. We used Keras’ Embedding layer to represent characters with 128-dimensional dense vectors. The input for Davis data set consisted of (85,128) and (1200, 128) dimensional matrices for the compounds and proteins, respectively. We represented KIBA data set with a (100,128) dimensional matrix for the compounds and a (1000, 128) dimensional matrix for the proteins.

## Experiments and Results

Here, we propose a novel drug - target binding affinity prediction method **based on only sequence information of compounds and proteins**. We utilized the Concordance Index (CI) to measure the performance of the proposed model and compared it with the current state-of-art methods that we chose as our baselines, namely a Kronecker Regularized Least Squares (KronRLS) based approach [47] and SimBoost [29]. We provide more information about these baseline methodologies, our model and experimental setup, as well as our results in the following subsections.

### Baselines

#### Kron-RLS

KronRLS aims to minimize the following function, where  $f$  is the prediction function [47]:

$$J(f) = \sum_{i=1}^m (y_i - f(x_i))^2 + \lambda \|f\|_k^2 \quad (3)$$



$\|f\|_k^2$  is the norm of  $f$ , which is related to the kernel function  $k$ , and  $\lambda > 0$  is a regularization hyper-parameter defined by the user. A minimizer for Equation 3 can be defined as follows [34]:

$$f(x) = \sum_{i=1}^m a_i k(x, x_i) \quad (4)$$

where  $k$  is the kernel function. In order to represent compounds, they utilized a similarity matrix computed using Pubchem structure clustering server (Pubchem Sim)(<http://pubchem.ncbi.nlm.nih.gov>), a tool that utilizes single linkage for cluster and uses 2D properties of the compounds to measure their similarity. As for proteins, the Smith-Waterman algorithm was used to construct a protein similarity matrix [52].

### SimBoost

SimBoost is a gradient boosting machine based method that depends on the features constructed from drugs, targets and drug-target pairs [29]. The proposed methodology uses feature engineering to build three types of features: (i) object-based features that utilize occurrence statistics and pairwise similarity information of drugs and targets, (ii) network-based features such as neighbor statistics, network metrics (betweenness, closeness etc.), PageRank score, which are collected from the respective drug-drug and target-target networks (In a drug-drug network, drugs are represented as nodes and connected to each other if the similarity of these two drugs is above a user-defined threshold. The target-target network is constructed in a similar way.) (iii) network-based features that are collected from a heterogeneous network (drug-target network) where a node can either be a drug or target and the drug nodes and target nodes are connected to each other via binding affinity value. In addition to the network metrics, neighbor statistics and PageRank scores, as well as latent vectors from matrix factorization are also included in this type of network.

These features are fed into a supervised learning method named gradient boosting regression trees [10,11] derived from gradient boosting machine model [21]. With gradient boosting regression trees, for a given drug-target pair  $dt_i$ , the binding affinity score  $\bar{y}$  predicted as follows [29]:

$$\bar{y}_i = \theta(dt_i) = \sum_{m=1}^M f_m(dt_i), f_m \in F \quad (5)$$

in which  $M$  denotes the number of regression trees and  $F$  represents the space of all possible trees. A regularized objective function to learn the set of trees  $f_m$  is described in the following form [29]:

$$R(\theta) = \sum_i l(y_i, \bar{y}_i) + \sum_m \alpha(f_m) \quad (6)$$

where  $l$  is the loss function that measures the difference between the actual binding affinity value  $y_i$  and the predicted value  $\bar{y}_i$ , while  $\alpha$  is the tuning parameter that controls the complexity of the model. The details are described in [10,11,29]. Similar to [47], [29] also used PubChem clustering server for drug similarity and Smith-Waterman for protein similarity computation.

## Evaluation Metrics

To evaluate the performance of a model that outputs continuous values, Concordance Index (CI) was used [26]:

$$CI = \frac{1}{Z} \sum_{\delta_i > \delta_j} h(b_i - b_j) \quad (7)$$

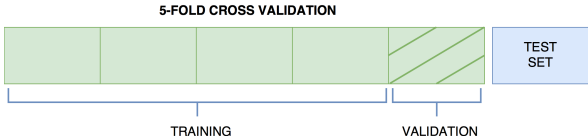
where  $b_i$  is the prediction value for the larger affinity  $\delta_i$ ,  $b_j$  is the prediction value for the smaller affinity  $\delta_j$ ,  $Z$  is a normalization constant,  $h(m)$  is the step function [47]:

$$h(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0.5, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (8)$$

The metric measures whether the predicted binding affinity values of two random drug-target pairs were predicted in the same order as their true values were. We used paired-t test for the statistical significance tests with 95% confidence interval. We also used MSE, which was explained in Section , as an evaluation metric.

## Experiment Setup

We evaluated the performance of the proposed model on the **benchmark data sets** [17, 54] similarly to [29]. They used nested-cross validation to decide on the best parameters for each test set. In order to learn a generalized model, we randomly divided our data set into six equal parts in which one part is selected as the independent test set. The remaining parts of the data set were used to determine the hyper-parameters via five-fold cross validation. Figure 3 illustrates the partitioning of the data set. The same setting with the same train and test folds was used for KronRLS [47] and Simboost [29] for a fair comparison.



**Figure 3.** Experiment setup.

We decided on three hyper-parameters for our model, namely the number of the filters (same for proteins and compounds), the length of the filter size for compounds, and the length of the filter size for proteins. We opted to experiment with different filter lengths for compounds and proteins instead of a common length, due to the fact that they have different alphabets. The hyper-parameter combination that provided the best average CI score over the validation set was selected as the best combination in order to model the test set. We first experimented with hyper-parameters chosen from a wide range and then fine-tuned the model. For example, to determine the number of filters we performed a search over [16, 32, 64, 128, 512]. We then narrowed the search range around the best performing parameter (e.g. if 16 was chosen as the best parameter, then our range was updated as [4, 8, 16, 20] etc.).

As explained in the Proposed Model subsection, the second convolution layer was set to contain twice the number of filters of the first layer, and the third one was set to contain

three times the number of filters of the first layer. 32 filters gave the best results over the cross-validation experiments. Therefore, in the final model, each CNN block consisted of three 1D convolutions of 32, 64, 96 filters. For all test results reported in Table 3 we used the same structure summarized in Table 2 except for the lengths of the pre-fine-tuned filters that were used for the compound CNN-block and protein CNN-block.

**Table 2.** Parameter settings for CNN based DeepDTA model

Parameters	Range
Number of filters	32*1; 32*2; 32*3
Filter length (compounds)	[4,6,8]
Filter length (proteins)	[4,8,12]
epoch	100
hidden neurons	1024; 1024; 512
batch size	256
dropout	0.1
optimizer	Adam
learning rate (lr)	0.001

In order to provide a more robust performance measure, we evaluated the performance over the independent test set, when the model was trained with the learned parameters in Table 2 on the five training sets that we used in five-fold cross validation (note that the validation sets were not used). The final CI score was reported as the average of these five results. Keras [13] with Tensorflow [1] back-end was used as development framework. Our experiments were run on OpenSuse 13.2 (3.50GHz Intel(R) Xeon(R) and GeForce GTX 1070 (8GB)). The work was accelerated by running on GPU with cuDNN [12]. We provide the train and test folds of the data sets (<https://github.com/hkmztrk/DeepDTA/>).

## Results

In this study, we propose a deep-learning model that uses two CNN-blocks to learn representations for drugs and targets based on their sequences. As a baseline for comparison, the KronRLS algorithm and SimBoost methods that use similarity matrices for proteins and compounds as input were used. The Smith-Waterman (S-W) and Pubchem Sim algorithms were used to compute the pairwise similarities for the proteins and ligands, respectively. We then used these S-W and Pubchem Sim similarity scores as inputs to the FC part of our model (DeepDTA) to evaluate the model. Finally, we used three alternative combinations in learning the hidden patterns of the data and used this information as input to our DeepDTA model. The combinations were (i) learning only compound representation with a CNN block and using S-W similarity as protein representation, (ii) learning only protein sequence representation with a CNN block and using Pubchem Sim to describe compounds, and (iii) learning both protein representation and compound representations with a CNN block. We call the last combination used with DeepDTA the combined model.

Tables 3 and 4 report the average MSE and CI scores over the independent test set of the five models trained with the same parameters (shown in Table 2) using the five different training sets for Davis and KIBA data sets.

**Table 3.** The average CI and MSE scores of the test set trained on five different training sets for the Davis data set. The standard deviations are given in parenthesis.

	Proteins	Compounds	CI (std)	MSE
KronRLS [47]	Smith-Waterman	Pubchem Sim	0.871 (0.0008)	0.379
SimBoost [29]	Smith-Waterman	Pubchem Sim	0.872 (0.002)	0.282
DeepDTA	Smith-Waterman	Pubchem Sim	0.790 (0.009)	0.608
DeepDTA	CNN	Pubchem Sim	0.835 (0.005)	0.419
DeepDTA	Smith-Waterman	CNN	<b>0.886</b> (0.008)	0.420
DeepDTA	CNN	CNN	0.878 (0.004)	<b>0.261</b>

**Table 4.** The average CI and MSE scores of the test set trained on five different training sets for the KIBA data set. The standard deviations are given in parenthesis.

	Proteins	Compounds	CI (std)	MSE
KronRLS [47]	Smith-Waterman	Pubchem Sim	0.782 (0.0009)	0.411
SimBoost [29]	Smith-Waterman	Pubchem Sim	0.836 (0.001)	0.222
DeepDTA	Smith-Waterman	Pubchem Sim	0.710 (0.002)	0.502
DeepDTA	CNN	Pubchem Sim	0.718 (0.004)	0.571
DeepDTA	Smith-Waterman	CNN	0.854 (0.001)	0.204
DeepDTA	CNN	CNN	<b>0.863</b> (0.002)	<b>0.194</b>

In the Davis data set, SimBoost and KronRLS methods perform similarly while the CI values for SimBoost is higher than that for KronRLS in the larger KIBA dataset. When the similarity measures S-W, for proteins, and Pubchem Sim, for compounds, are used with the the fully-connected part of the neural networks (DeepDTA), the CI drops to 0.79 for the Davis data set and to 0.71 for the KIBA data set. The MSE increases to more than 0.5. These results suggest that the use of a feed-forward neural network with predefined features is not sufficient to describe drug target interactions and to predict drug target affinities. Therefore, we used CNN layers to learn representations of drugs and proteins to capture hidden patterns in the datasets.

We first used CNN to learn representations of proteins and used the predefined Pubchem Sim scores for the ligands. Using this combination did not improve the results suggesting that use of a CNN architecture is not effective enough to learn from amino acid sequences.

Then we used the CNN block to learn compound representations from SMILES and used the predefined S-W scores for the proteins. This combination outperformed the baselines on the KIBA data set with statistical significance (p-value of 0.0001 for both SimBoost and KronRLS), and on the Davis data set (p-value of around 0.03 for both SimBoost and KronRLS). These results suggested that the CNN is able to capture more information than Pubchem Sim in the compound representation task.

Motivated by this result, we tested the combined CNN model in which both protein and compound representations are learned from the CNN layer. This method performed as well as the baseline methods with CI score of 0.878 on the Davis data set and achieved the best CI score (0.863) on the KIBA data set with statistical significance over both baselines (p-value of 0.0001 for both). The MSE values of this model were also notably lower than the MSE of the baseline models on both data sets. Even though learning protein representations with CNN was not effective, combination of the two CNN blocks for proteins and ligands provided a strong model.

## Conclusion

We propose a deep-learning based approach to predict drug-target binding affinity using only sequences of proteins and drugs. We use Convolutional Neural Networks (CNN) to learn representations from the raw sequence data of proteins and drugs and fully connected layers (DeepDTA) in the affinity prediction task. We compare the performance of the proposed model with two recent studies that employed the KronRLS regression algorithm [47] and the SimBoost method [29] as our baselines. We perform our experiments on the Davis kinase - drug data set and the KIBA data set.

Our results showed that the use of predefined features with DeepDTA is not sufficient to describe protein - ligand interactions. However, when two CNN-blocks that learn representations of proteins and drugs based on raw sequence data are used in conjunction with DeepDTA, the performance increases significantly compared to both baseline methodologies for both KIBA and Davis data sets. Furthermore, the model that uses CNN to learn compound representations from SMILES and S-W similarities of proteins also achieves better performance than the baselines.

We observed that the model that uses CNN-block to learn proteins and 2D compound similarity to represent compounds performed poorly compared to the other methods that employ CNN. This might be an indication that amino-acids require a structure that can handle their ordered relationships, which the CNN architecture failed to capture successfully. Long-Short Term Memory (LSTM), which is a special type of Recurrent Neural Networks (RNN), could be a more suitable approach to learn from protein sequences, since the architecture has memory blocks that allow effective learning from a long sequence. LSTM architecture has been successfully employed to tasks such as detecting homology [31], constructive peptide design [41] and function prediction [39] that utilize amino-acid sequences. As future work, we also aim to utilize a recent ligand-based protein representation method proposed by our team that uses SMILES sequences of the interacting ligands to describe proteins [46] .

The results indicated that deep-learning based methodologies performed notably better than the baseline methods with a statistical significance when the data set grows in size, as the KIBA data set is four times larger than the Davis data set. The improvement over the baseline was significantly higher for the KIBA data set (from CI score of 0.836 to 0.863) compared to the Davis data set (from CI score of 0.872 to 0.878). The increase in the data enables the deep learning architectures to capture the hidden information better.

The major contribution of this study is the presentation of a novel deep learning-based model for drug - target affinity prediction that uses only character representations of proteins and drugs. By simply using raw sequence information for both drugs and targets, we were able to achieve similar or better performance than the baseline methods that depend on multiple different tools and algorithms to extract features.

A large percentage of proteins remains untargeted, either due to bias in the drug discovery field for a select group of proteins or due to their undruggability, and this untapped pool of proteins has gained interest with protein deorphanizing efforts [19,20,43]. As future work, we will focus on building an effective representation for protein sequences. The methodology can then be extended to predict the affinity of known compounds/targets to novel targets/drugs as well as to the prediction of the affinity of novel drug-target pairs.

## Acknowledgments

TUBITAK-BIDEB 2211-E Scholarship Program (to HO) and BAGEP Award of the Science Academy (to AO) are gratefully acknowledged. We thank Ethem Alpaydın, Attila Gürsoy and Pınar Yolum for the helpful discussions.

## Funding

This work is funded by Bogazici University Research Fund (BAP) Grant Number 12304.

## References

1. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
2. R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl\_1):D115–D119, 2004.
3. P. J. Ballester and J. B. Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.
4. K. Bleakley and Y. Yamanishi. Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*, 25(18):2397–2403, 2009.
5. E. E. Bolton, Y. Wang, P. A. Thiessen, and S. H. Bryant. Pubchem: integrated platform of small molecules and biological activities. *Annual reports in computational chemistry*, 4:217–241, 2008.
6. D.-S. Cao, S. Liu, Q.-S. Xu, H.-M. Lu, J.-H. Huang, Q.-N. Hu, and Y.-Z. Liang. Large-scale prediction of drug–target interactions using protein sequences and drug topological structures. *Analytica chimica acta*, 752:1–10, 2012.
7. D.-S. Cao, L.-X. Zhang, G.-S. Tan, Z. Xiang, W.-B. Zeng, Q.-S. Xu, and A. F. Chen. Computational prediction of drug- target interactions using chemical, biological, and network features. *Molecular Informatics*, 33(10):669–681, 2014.
8. R. Z. Cer, U. Mudunuri, R. Stephens, and F. J. Lebeda. Ic 50-to-k i: a web-based tool for converting ic 50 to k i values for inhibitors of enzyme activity and ligand binding. *Nucleic acids research*, 37(suppl\_2):W441–W445, 2009.
9. K. C. Chan, Z.-H. You, et al. Large-scale prediction of drug-target interactions from deep representations. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 1236–1243. IEEE, 2016.
10. T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

11. T. Chen and T. He. Higgs boson discovery with boosted trees. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pages 69–80, 2015.
12. S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shenhav. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.
13. F. Chollet et al. Keras, 2015.
14. D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.
15. 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*, 53(12):3399–3409, 2013.
16. G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.
17. M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.
18. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.
19. A. M. Edwards, R. Isserlin, G. D. Bader, S. V. Frye, T. M. Willson, and H. Y. Frank. Too many roads not taken. *Nature*, 470(7333):163, 2011.
20. O. Fedorov, S. Müller, and S. Knapp. The (un) targeted cancer kinome. *Nature chemical biology*, 6(3):166, 2010.
21. J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
22. J. Gabel, J. Desaphy, and D. Rognan. Beware of machine learning-based scoring functions on the danger of developing black boxes. *Journal of chemical information and modeling*, 54(10):2807–2815, 2014.
23. J. Gomes, B. Ramsundar, E. N. Feinberg, and V. S. Pande. Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603*, 2017.
24. R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint arXiv:1610.02415*, 2016.

25. M. Gönen. Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310, 2012.
26. M. Gönen and G. Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.
27. A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.
28. M. Hamanaka, K. Taneishi, H. Iwata, J. Ye, J. Pei, J. Hou, and Y. Okuno. Cgbvs-dnn: Prediction of compound-protein interactions based on deep learning. *Molecular Informatics*, 2016.
29. 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*, 9(1):24, 2017.
30. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.
31. S. Hochreiter, M. Heusel, and K. Obermayer. Fast model-based protein homology detection without alignment. *Bioinformatics*, 23(14):1728–1736, 2007.
32. S. Jastrzkeski, D. Lesniak, and W. M. Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.
33. L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740, 2014.
34. G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971.
35. D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
36. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
37. M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, 2014.
38. H. Li, K.-S. Leung, M.-H. Wong, and P. J. Ballester. Low-quality structural and interaction data improves binding affinity prediction via random forest. *Molecules*, 20(6):10947–10962, 2015.
39. X. Liu. Deep recurrent neural network for protein function prediction from sequence. *arXiv preprint arXiv:1701.08318*, 2017.



40. J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.
41. A. T. Muller, J. A. Hiss, and G. Schneider. Recurrent neural network model for constructive peptide design. *Journal of Chemical Information and Modeling*, 2018.
42. V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
43. M. J. O’Meara, S. Ballouz, B. K. Shoichet, and J. Gillis. Ligand similarity complements sequence, physical interaction, and co-expression for gene function prediction. *PloS one*, 11(7):e0160098, 2016.
44. T. Oprea and J. Mestres. Drug repurposing: far beyond new targets for old drugs. *The AAPS journal*, 14(4):759–763, 2012.
45. H. Öztürk, E. Ozkirimli, and A. Özgür. A comparative study of smiles-based compound similarity functions for drug-target interaction prediction. *BMC bioinformatics*, 17(1):128, 2016.
46. H. Öztürk, E. Ozkirimli, and A. Özgür. A novel methodology on distributed representations of proteins using their interacting ligands. *Bioinformatics, accepted for publication*, 2018.
47. T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang, and T. Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, page bbu010, 2014.
48. M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, and D. R. Koes. Protein–ligand scoring with convolutional neural networks. *J. Chem. Inf. Model*, 57(4):942–957, 2017.
49. P. W. Rose, A. Prlić, A. Altunkaya, C. Bi, A. R. Bradley, C. H. Christie, L. D. Costanzo, J. M. Duarte, S. Dutta, Z. Feng, et al. The rcsb protein data bank: integrative view of protein, gene and 3d structural information. *Nucleic acids research*, page gkw1000, 2016.
50. P. A. Shar, W. Tao, S. Gao, C. Huang, B. Li, W. Zhang, M. Shahan, C. Zheng, Y. Bai, and Y. Wang. Pred-binding: large-scale protein–ligand binding affinity prediction. *Journal of enzyme inhibition and medicinal chemistry*, 31(6):1443–1450, 2016.
51. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
52. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.
53. N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

54. J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg, and T. Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.
55. K. Tian, M. Shao, S. Zhou, and J. Guan. Boosting compound-protein interaction prediction by deep learning. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 29–34. IEEE, 2015.
56. T. van Laarhoven, S. B. Nabuurs, and E. Marchiori. Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, 2011.
57. I. Wallach, M. Dzamba, and A. Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.
58. L. Wang, Z.-H. You, X. Chen, S.-X. Xia, F. Liu, X. Yan, Y. Zhou, and K.-J. Song. A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network. *Journal of Computational Biology*, 2017.
59. M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, and H. Lu. Deep-learning-based drug–target interaction prediction. *Journal of Proteome Research*, 16(4):1401–1409, 2017.
60. H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218):1254806, 2015.
61. 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*, 24(13):i232–i240, 2008.