predicting drug-target binding affinity with graph neural networks

Presented by: Youssef Ezz Eldeen Ezzat

Directed By: Francesc Seratosa

Contents

- Introduction
- Paper
- Data representation
- Model
- Results

Introduction

- A virus encodes one or more proteases which are enzymes that spur the formation of new protein products, thus play crucial roles in virus replication
- proteases are important targets for the design and development of potent antiviral agents or drugs

Introduction

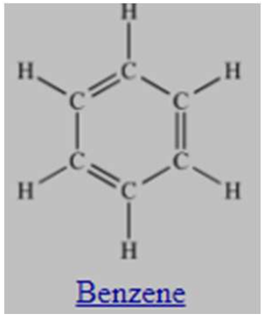
- Binding affinity is the strength of the binding interaction between a single molecule (e.g., a virus protein) to its ligand or binding partner (e.g., a drug)

featurizing drug molecule

- In order to perform machine learning on molecules, we need to transform them into feature vectors that can be used as inputs to models
  - SMILES notation
  - Molecular graph

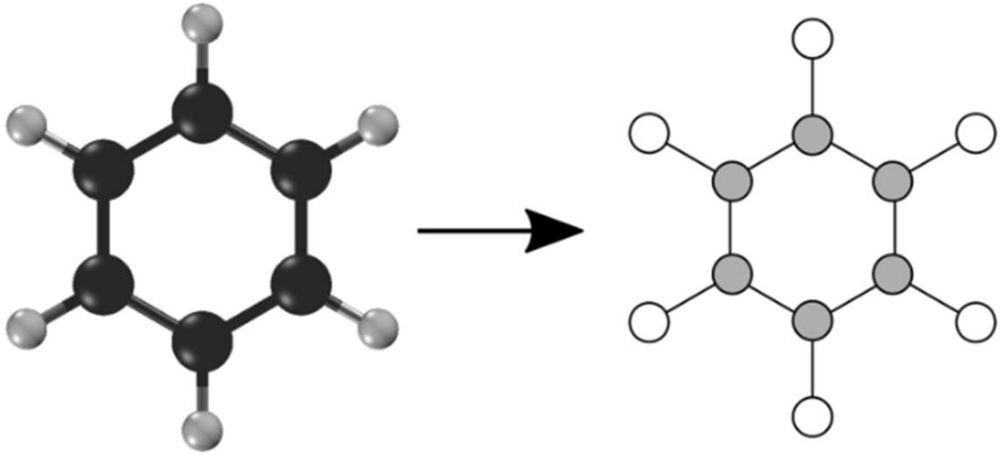
SMILES notation

- “Simplified Molecular-Input Line-Entry System”
- popular method for specifying molecules with text strings.
- invented to represent molecules to be readable by humans and computers
  - Methane: "C"
  - Ethanol: "CCO"
  - Benzene: "c1ccccc1“
  - Glucose: "OC[C@@H]1OC@HC@@HC@H[C@H]1O"



molecular graph

- A molecular graph describes the set of atoms in a molecule and how they are bonded together
- G = (V,E), where V is the set of N nodes and E is the set of edges represented as an adjacency matrix A



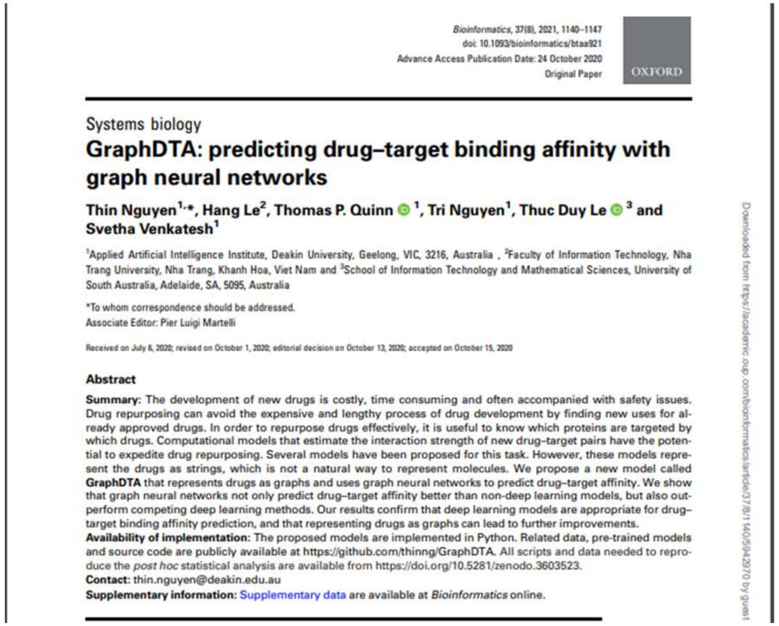
An example of converting a benzene molecule into a molecular graph. Note that atoms are converted into nodes and chemical bonds into edges.

Previous Work

- collaborative filtering (2017): the SimBoost model uses the affinity similarities among drugs and among targets to build new features.
- DeepDTA model (2018): uses 1D representations and layers of 1D convolutions (with pooling) to capture predictive patterns within the data
- WideDTA model (2019):extension of DeepDTA in which the sequences of the drugs and proteins are first summarized as higher-order features

GraphDTA paper overview

- a new neural network architecture capable of directly modeling drugs as molecular graphs
- outperforms previous deep learning models.
- directly modeling drugs as molecular graphs



Bioinformatics, 37(8), 2021, 1140–1147
doi: 10.1093/bioinformatics/btaa921
Advance Access Publication Date: 24 October 2020
Original Paper

Systems biology

GraphDTA: predicting drug–target binding affinity with graph neural networks

Thin Nguyen[1,*], Hang Le[2], Thomas P. Quinn [1], Tri Nguyen[1], Thuc Duy Le [3] and Svetha Venkatesh[1]

[1]Applied Artificial Intelligence Institute, Deakin University, Geelong, VIC, 3216, Australia , [2]Faculty of Information Technology, Nha Trang University, Nha Trang, Khanh Hoa, Viet Nam and [3]School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA, 5095, Australia

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on July 6, 2020; revised on October 1, 2020; editorial decision on October 13, 2020; accepted on October 15, 2020

Abstract

Summary: The development of new drugs is costly, time consuming and often accompanied with safety issues. Drug repurposing can avoid the expensive and lengthy process of drug development by finding new uses for already approved drugs. In order to repurpose drugs effectively, it is useful to know which proteins are targeted by which drugs. Computational models that estimate the interaction strength of new drug–target pairs have the potential to expedite drug repurposing. Several models have been proposed for this task. However, these models represent the drugs as strings, which is not a natural way to represent molecules. We propose a new model called **GraphDTA** that represents drugs as graphs and uses graph neural networks to predict drug–target affinity. We show that graph neural networks not only predict drug–target affinity better than non-deep learning models, but also outperform competing deep learning methods. Our results confirm that deep learning models are appropriate for drug–target binding affinity prediction, and that representing drugs as graphs can lead to further improvements.

Availability of implementation: The proposed models are implemented in Python. Related data, pre-trained models and source code are publicly available at https://github.com/thinng/GraphDTA. All scripts and data needed to reproduce the *post hoc* statistical analysis are available from https://doi.org/10.5281/zenodo.3603523.

Contact: thin.nguyen@deakin.edu.au

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Bioinformatics, Volume 37, Issue 8, March 2021, Pages 1140–1147, https://doi.org/10.1093/bioinformatics/btaa921

binding affinity measures

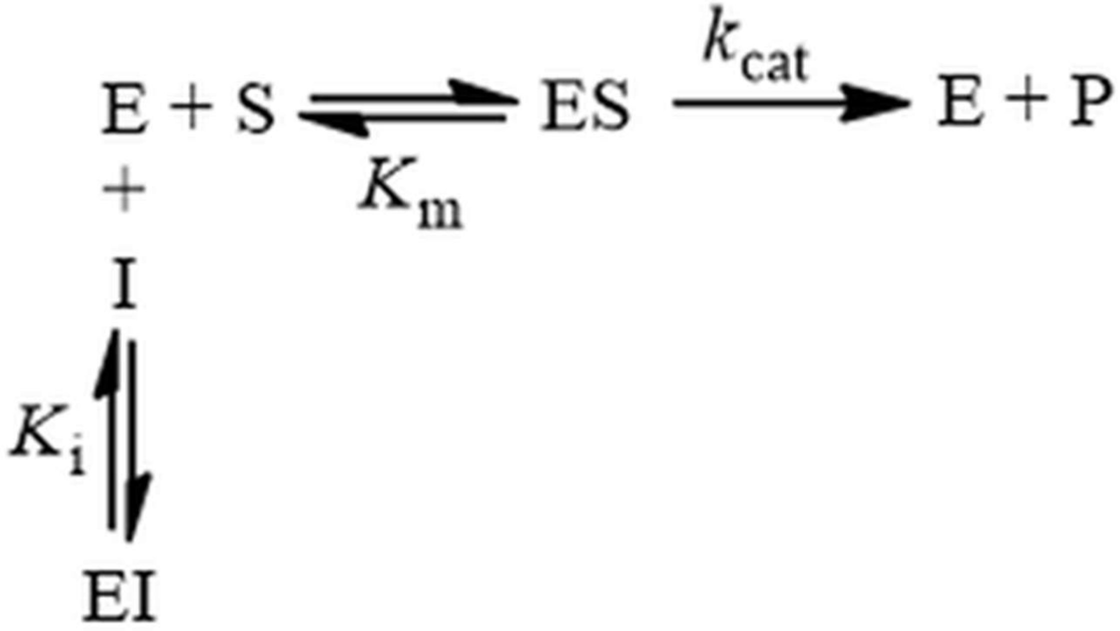
- The kinase dissociation constant(Kd)
  - measures the equilibrium between the ligand(drug)-protein complex and the dissociated components

$$\mathrm{PL} \underset{K_\mathrm{d}}{\rightleftharpoons} \mathrm{P} + \mathrm{L} \qquad K_\mathrm{d} = \frac{[P][L]}{[PL]}$$

  - Where [P] is the free protein concentration
  - [L] is the free ligand concentration
  - [PL] is the protein-ligand complex

binding affinity measures

- The kinase Inhibition Constant(Ki)
  - represents the affinity of the drug molecule for its target receptor, specifically in the context of competitive inhibition.



binding affinity measures

- inhibitory concentration 50% (IC50)
  - the concentration at which the inhibitor causes a 50% inhibition of enzymatic activity
  - less precise than Ki or Kd
  - A lower IC50 value indicates a higher affinity of the drug for the receptor

$$0.5 = \frac{K_\mathrm{m} + [S]}{K_\mathrm{m}\left(1 + \frac{\mathrm{IC}_{50}}{K_\mathrm{i}}\right) + [S]} \qquad \mathrm{IC}_{50} = K_\mathrm{i}\left(1 + \frac{[S]}{K_\mathrm{m}}\right)$$

  - [S] is the concentration of the natural substrate that competes with the inhibitor for binding to the target.

Bioactivity values found from ChEMBL for the imatinib-SRC pair

Drug	Type	Value	Units	Target
IMATINIB	Ki	31000	nM	SRC
IMATINIB	Kd	10000	nM	SRC
IMATINIB	IC50	100000	nM	SRC

Making Sense of Large-Scale Kinase Inhibitor Bioactivity Data Sets: A Comparative and Integrative Analysis
, https://pubs.acs.org/doi/10.1021/ci400709d

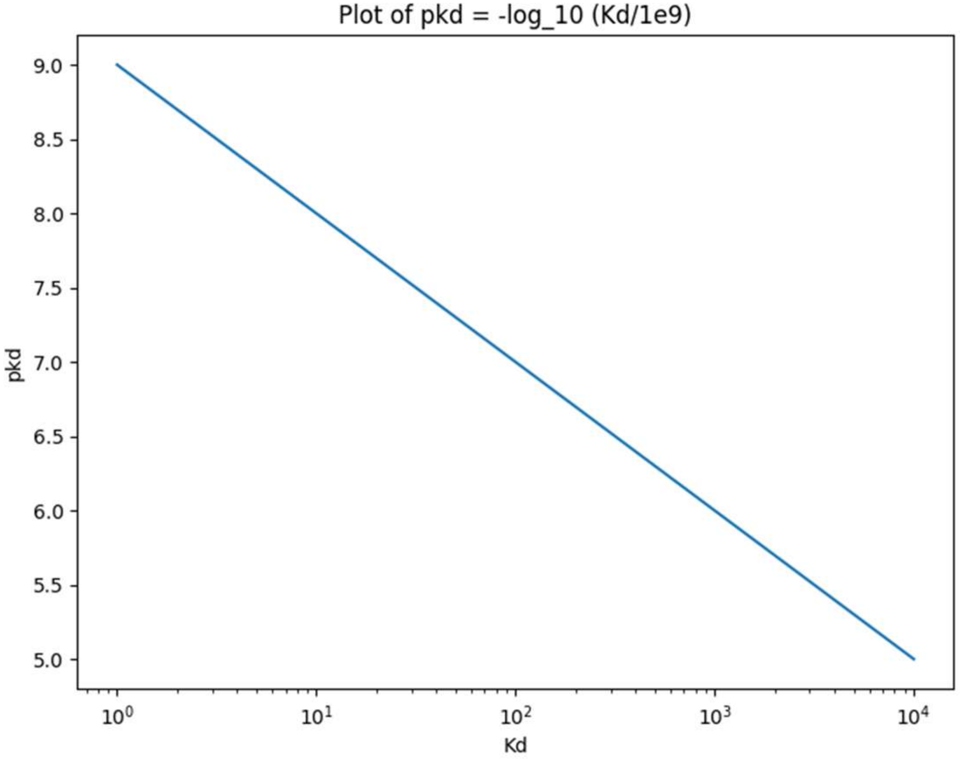
Datasets

- Benchmark dataset davis
- Benchmark dataset kiba
- In house dataset URV

Datasets

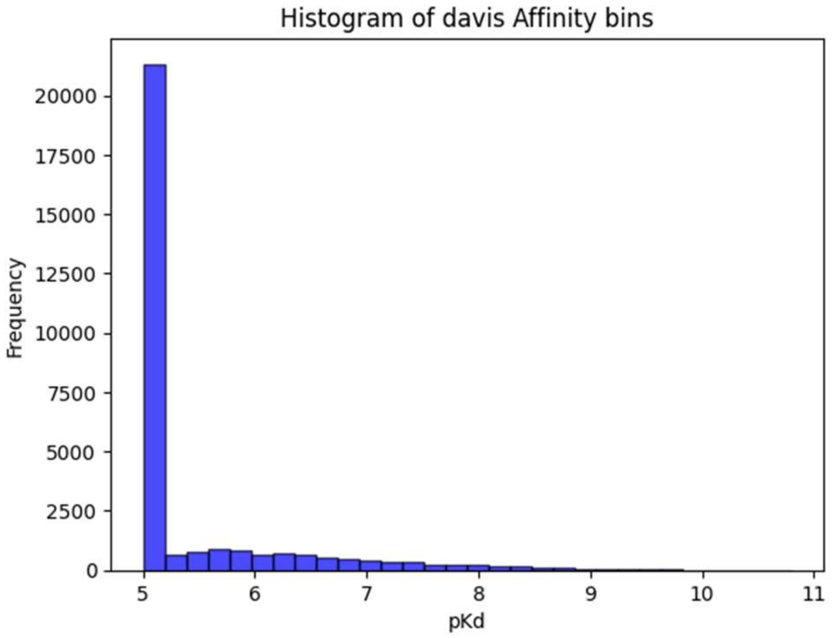
- Benchmark dataset davis
  - Kd values in the Davis dataset were transformed into logspace (pKd) as: $\boldsymbol{pkd = -log_{10}(^{Kd}/_{1e9})}$
  - ranging from 5.0 to 10.8

Plot of pkd = -log_10 (Kd/1e9)



Datasets

- Benchmark dataset davis
  - contains the binding affinities for all pairs of 68 drugs and 442 targets, total of 30056 interactions
  - 69% of which have affinity values of 10000 nM (pKd=5) indicating weak or no interaction.



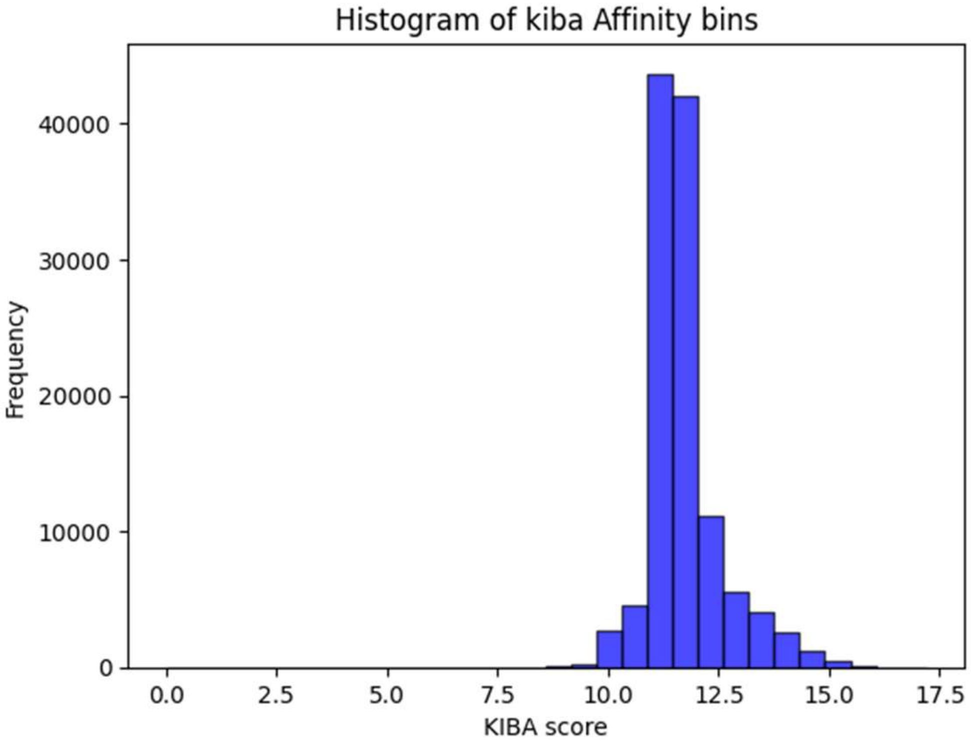
Datasets

- Benchmark dataset kiba
  - Kinase Inhibitor Bioactivity Data Set
  - binding affinity might be measured by Kd, Ki or IC50
  - integrates the information from IC50, Ki , and Kd measurements into a single bioactivity score

$$\mathrm{KIBA} = \begin{cases} K_i.\ \mathrm{adj} & \text{if } \mathrm{IC}_{50} \text{ and } K_i \\ & \quad \text{are present} \\ K_d.\ \mathrm{adj} & \text{if } \mathrm{IC}_{50} \text{ and } K_d \\ & \quad \text{are present} \\ (K_i.\ \mathrm{adj} + K_d.\ \mathrm{adj})/2 & \text{if } \mathrm{IC}_{50},\ K_i, \text{ and } K_d \\ & \quad \text{are present} \end{cases}$$

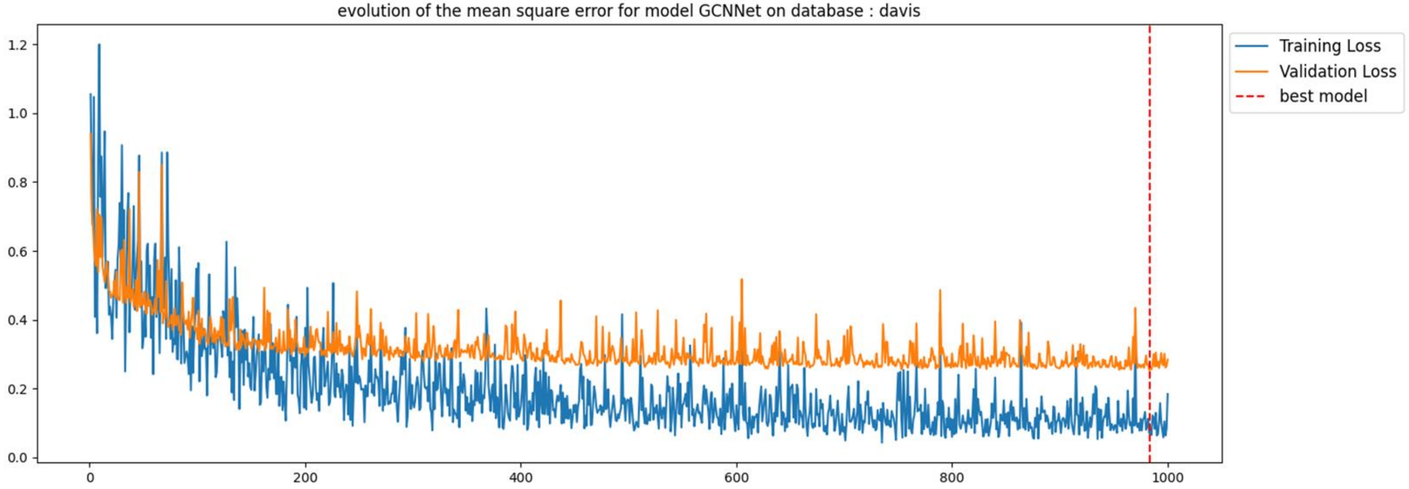
Datasets

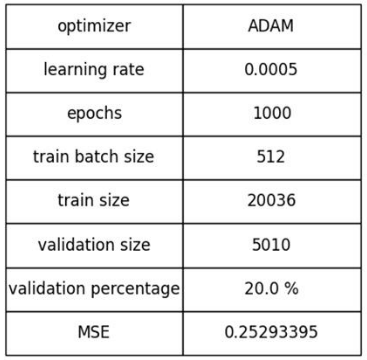
- Benchmark dataset kiba
  - measured as KIBA scores and ranging from 0.0 to 17.2
  - Total of most interactions between 10 and 15



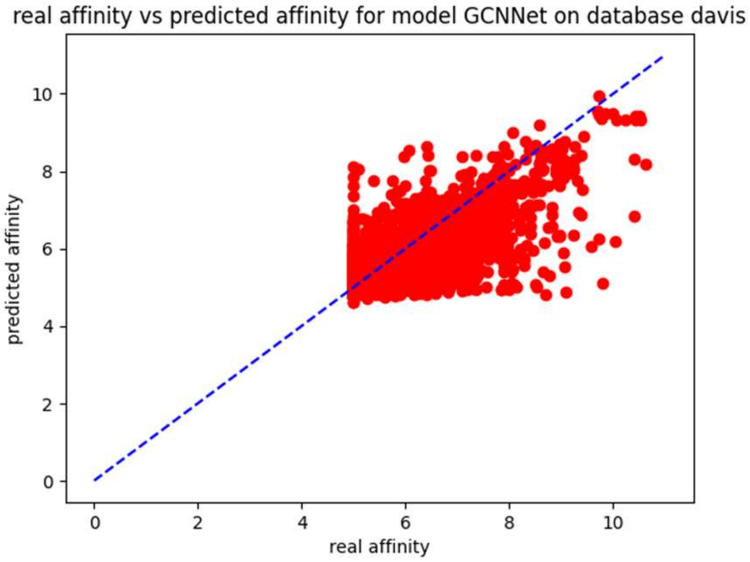
Results

- train and test GCN-based model with davis dataset



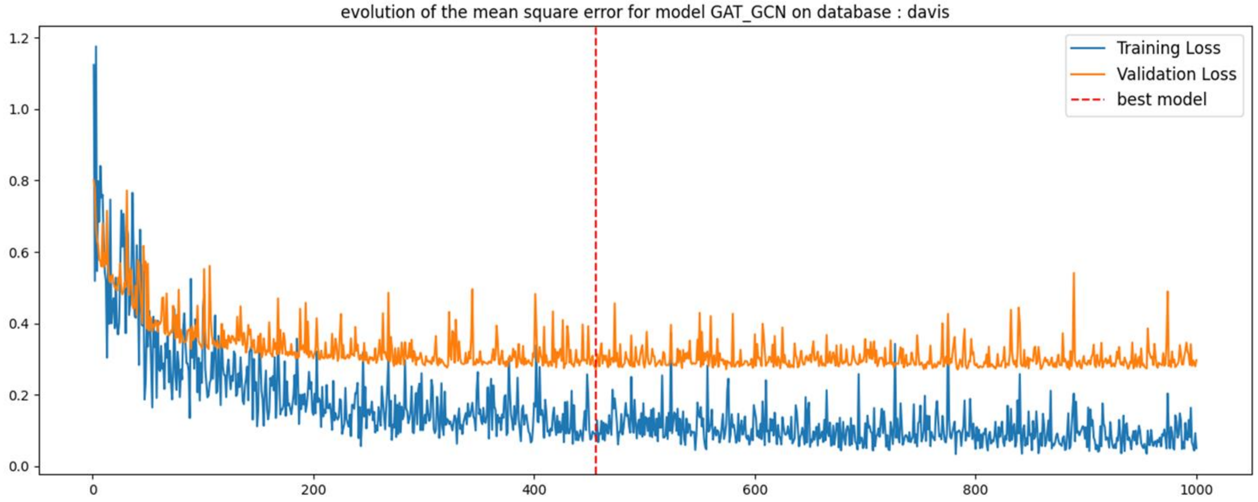


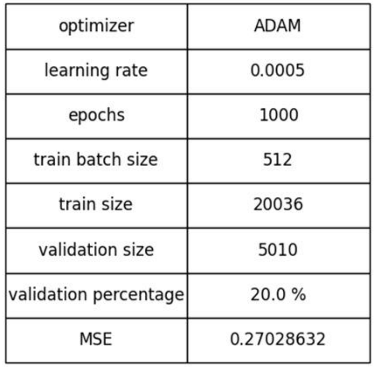
optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	20036
validation size	5010
validation percentage	20.0 %
MSE	0.25293395



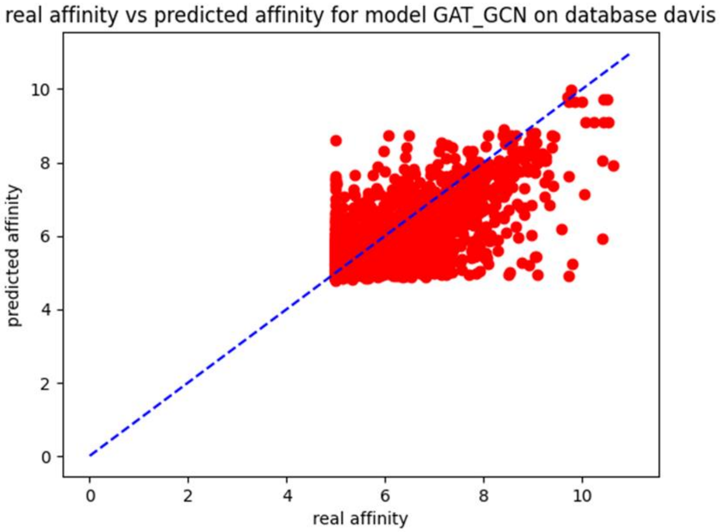
Results

- train and test GATGCN-based model with davis



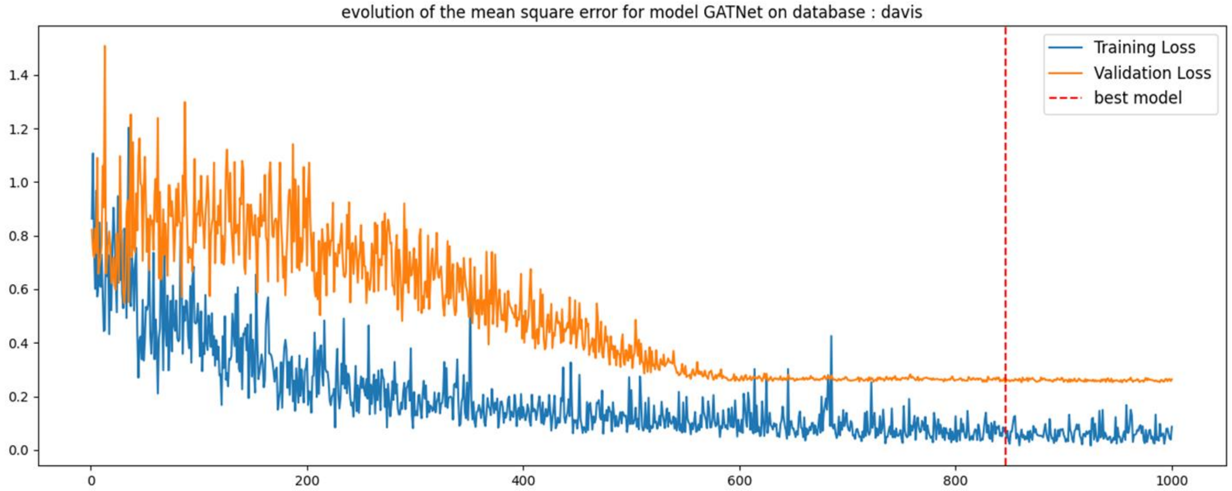


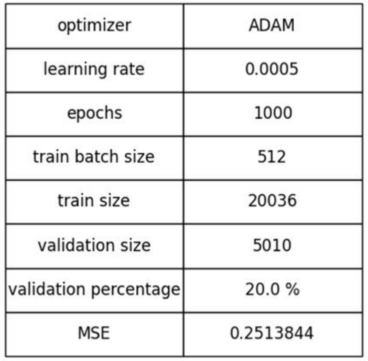
optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	20036
validation size	5010
validation percentage	20.0 %
MSE	0.27028632



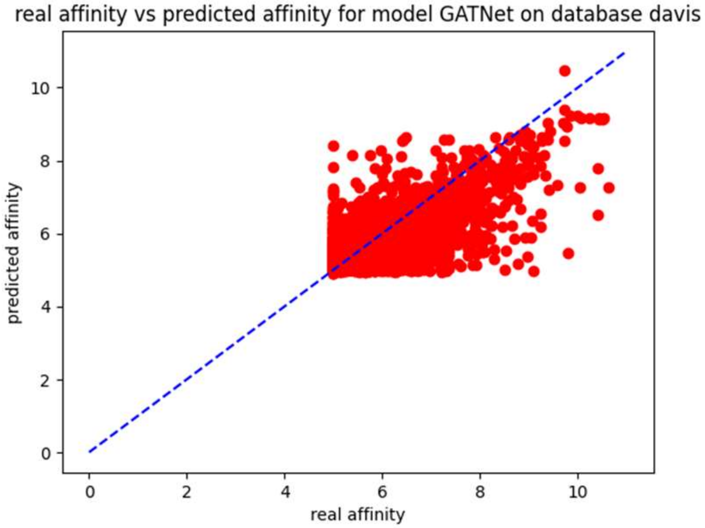
Results

- train and test GAT-based model with davis dataset


evolution of the mean square error for model GATNet on database : davis

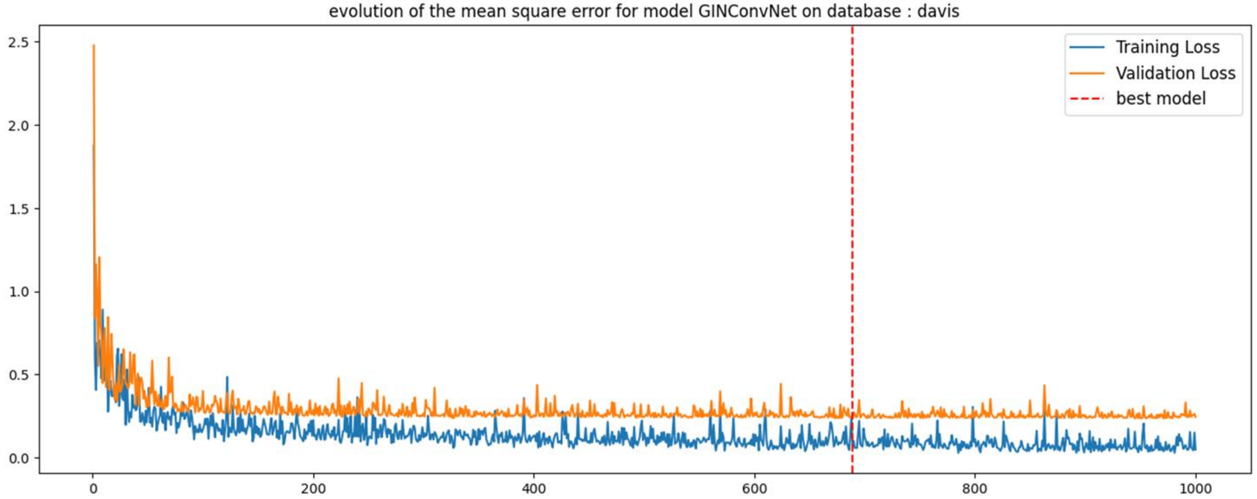


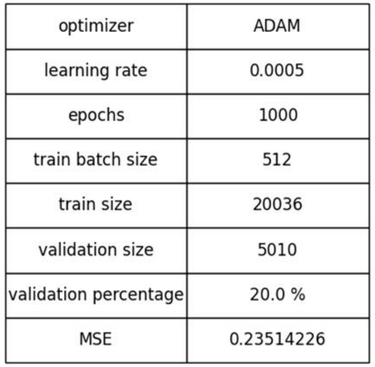
optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	20036
validation size	5010
validation percentage	20.0 %
MSE	0.2513844


real affinity vs predicted affinity for model GATNet on database davis

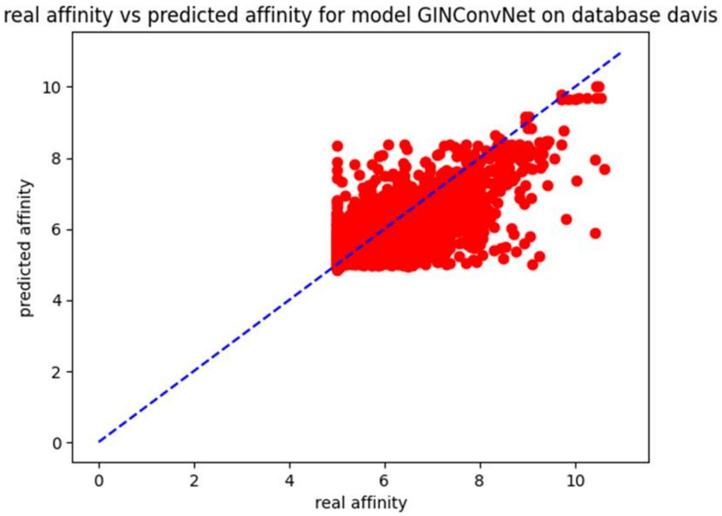
Results

- train and test GinConv-based model with davis


evolution of the mean square error for model GINConvNet on database : davis

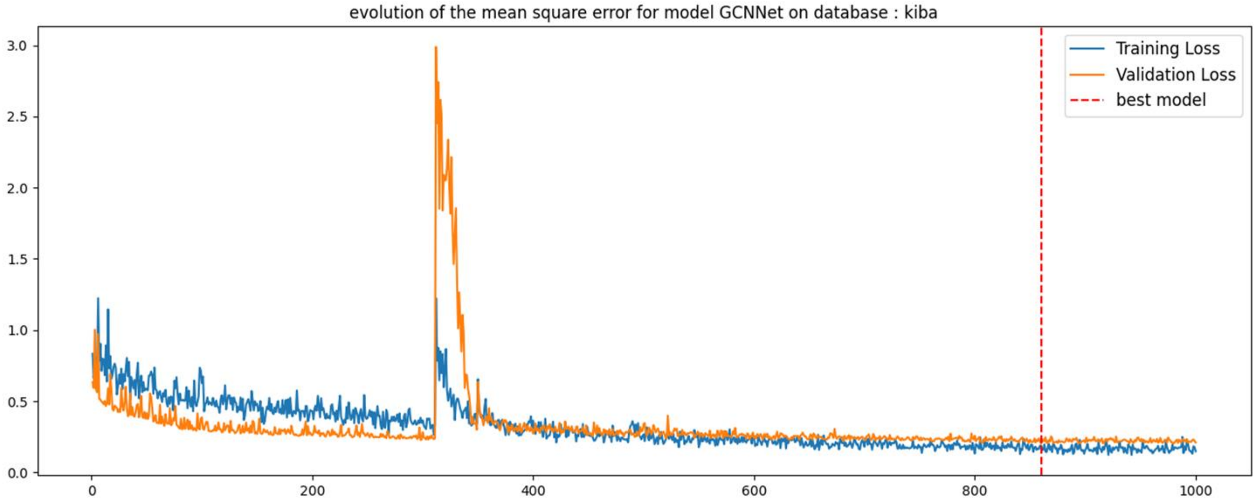


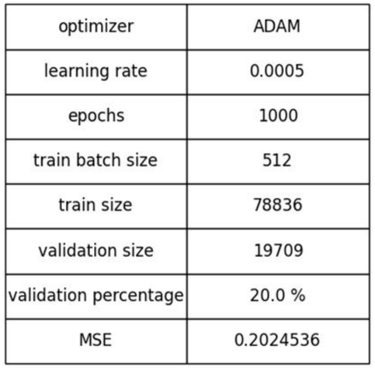
optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	20036
validation size	5010
validation percentage	20.0 %
MSE	0.23514226


real affinity vs predicted affinity for model GINConvNet on database davis

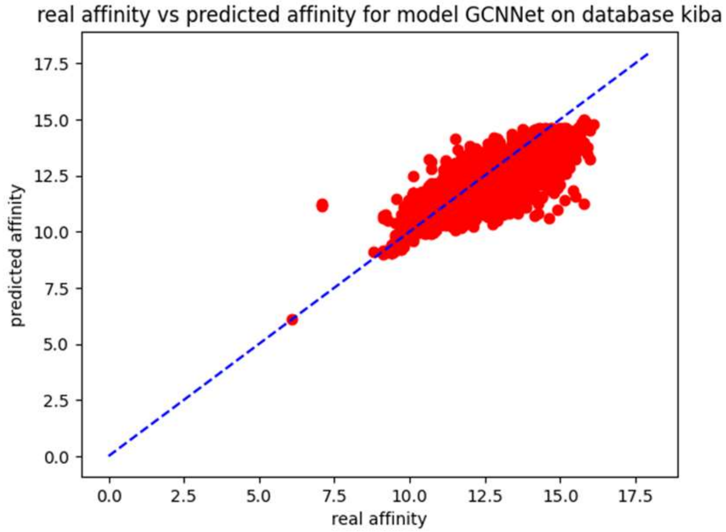
Results

- train and test GCN-based model with kiba dataset



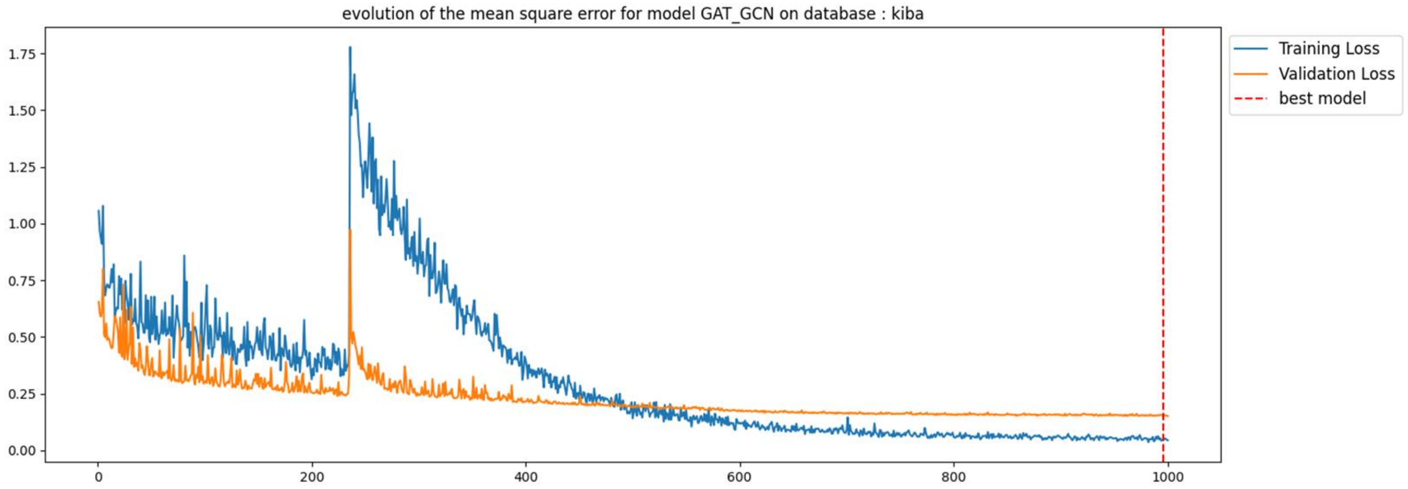


optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	78836
validation size	19709
validation percentage	20.0 %
MSE	0.2024536

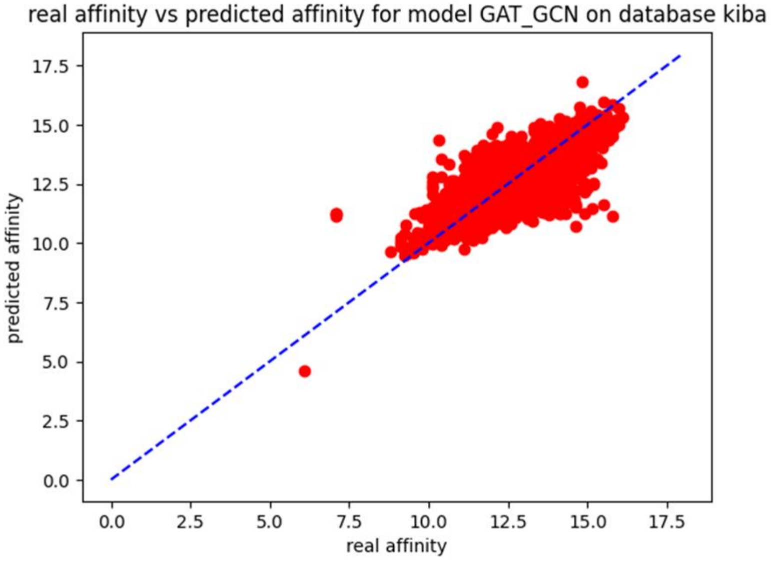


Results

- train and test GATGCN-based model with kiba

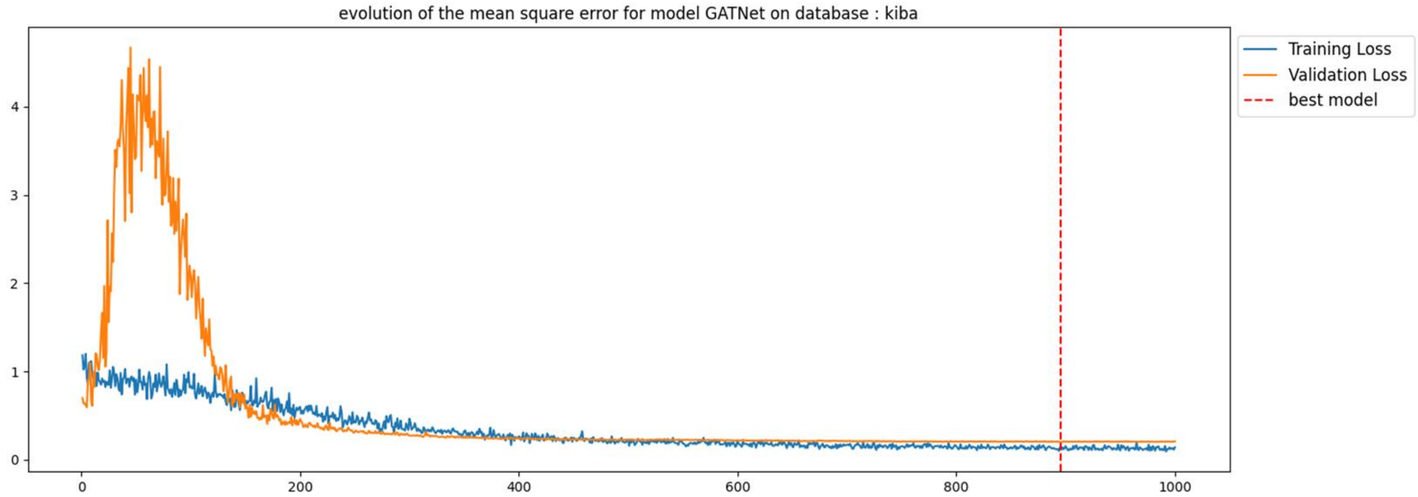


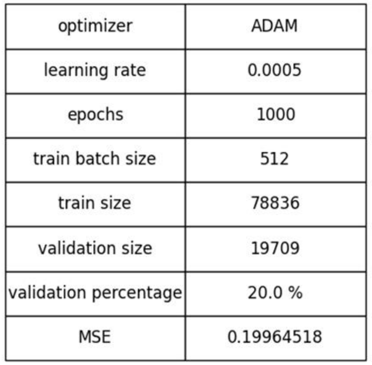
optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	78836
validation size	19709
validation percentage	20.0 %
MSE	0.15026996



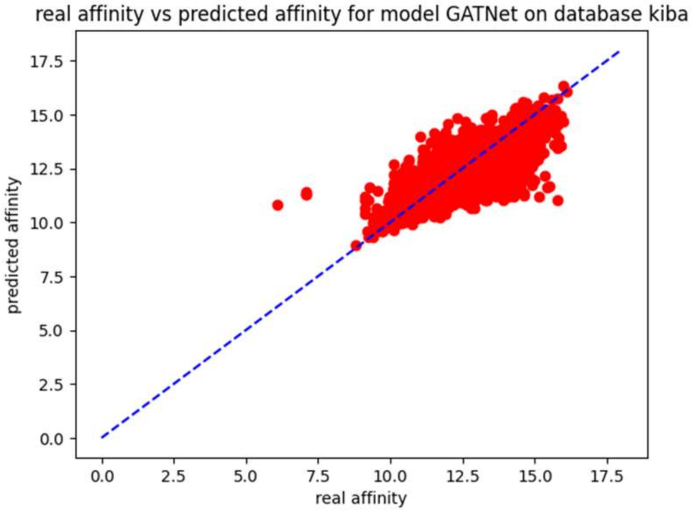
Results

- train and test GAT-based model with kiba dataset



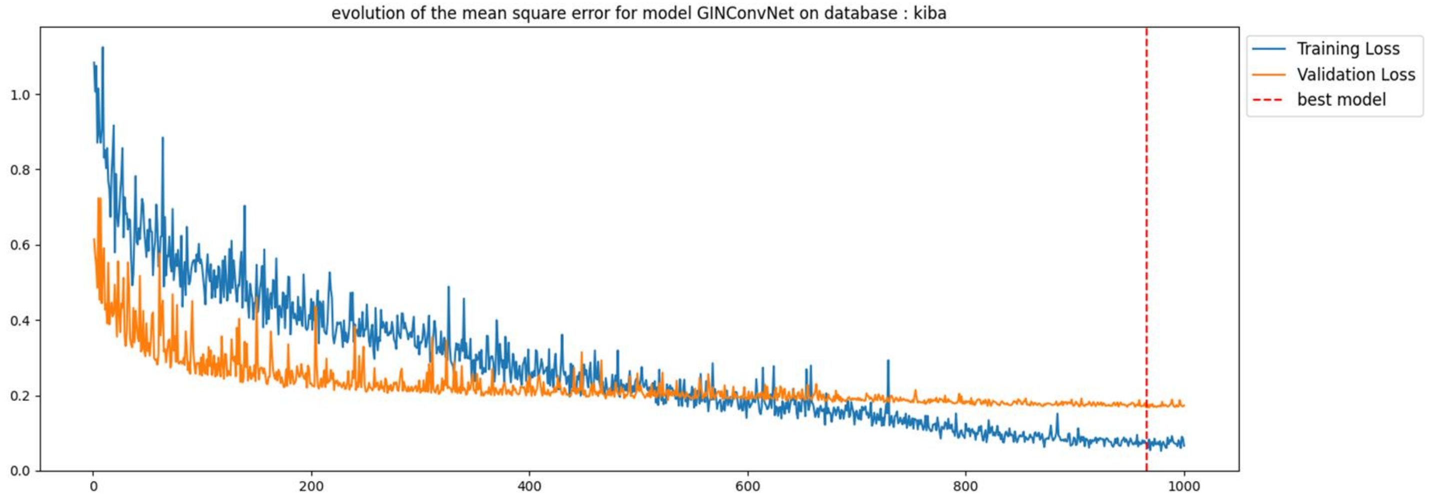


optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	78836
validation size	19709
validation percentage	20.0 %
MSE	0.19964518

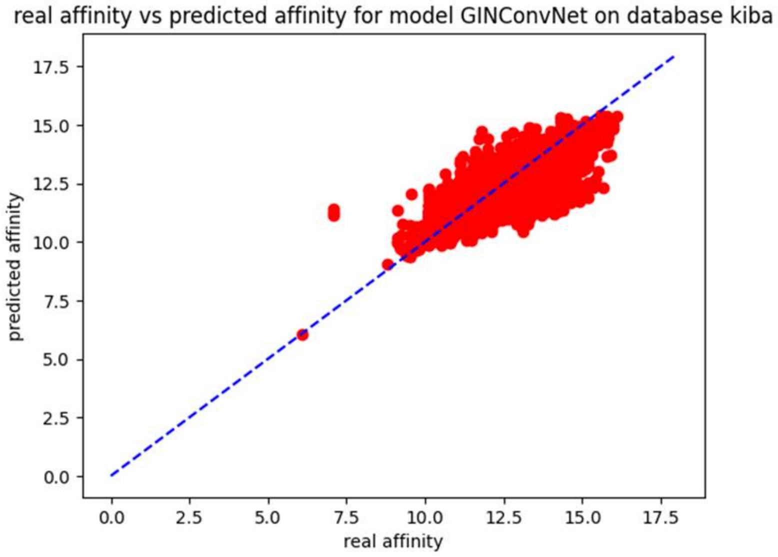


Results

- train and test GINCONV-based model with kiba



optimizer	ADAM
learning rate	0.0005
epochs	1000
train batch size	512
train size	78836
validation size	19709
validation percentage	20.0 %
MSE	0.1673416



Summary of paper results

model	dataset	MSE in paper	MSE obtained
GCN-based	davis	0.254	0.25
GATGCN-based	davis	0.245	0.27
GAT-based	davis	0.232	0.25
GINCONV-based	davis	0.229	0.24
GCN-based	kiba	0.179	0.2
GATGCN-based	kiba	0.147	0.15
GAT-based	kiba	0.139	0.2
GINCONV-based	kiba	0.139	0.17