# Materials and Methods

## Dataset Compilation

### Training Dataset

A training dataset of 19491 enzyme-substrate interactions between 3408 unique enzymes and 9572 unique substrates was compiled by parsing the BRENDA database (1). Among the 19491 interactions, there were 9789 positive interactions between enzymes and their natural substrates, and 9702 negative interactions created through a substrate similarity guided approach discussed in S1 Text.

### Validation Dataset

The validation dataset consisted of 6497 enzyme-substrate interactions between 2314 unique enzymes and 4846 unique substrates parsed from BRENDA, distinct from the training set interactions, among which 3205 were positive interactions whereas 3292 were negative. The negative interactions in validation dataset were created using the same technique as those in the training set.

### Test Dataset

An independent test set of 69409 positive enzyme-substrate interactions between 29429 unique enzymes and 2191 unique substrates was created by extracting enzyme-substrate pairs from the KEGG database (2). We used a different database to ensure there is no data leakage from training to test set which would eventually help us estimate model generalizability accurately.

## Feature Extraction

### Substrates

The substrates were numerically encoded by calculating their Morgan fingerprint with a radius of two and converting them into a 2048 dimensional binary vector using rdkit package (3). The Morgan fingerprint encoded substrates were used to train both Kernel Density Estimation and DeepConv-DTI models while Tanimoto similarity model used SMILES notation of the substrates to calculate similarity scores.

### Enzyme Sequences

Multiple feature representation methods were used to numerically encode the enzyme sequences. Tanimoto similarity model utilized the raw sequences to calculate pairwise similarity between enzymes whereas Dipeptide Deviation from Expected mean (4) sequence encoding technique was used to create features for training and testing the Kernel Density Estimation model. DeepConv-DTI on the other hand used built-in convolution filters to encode the enzyme sequences.

## Models

### Tanimoto similarity model

Tanimoto similarity model was used as a baseline to benchmark the performance of the other more complicated models. The model calculates similarity scores between a novel/test enzyme-substrate pair and existing/known enzyme substrate pairs and flags the pair as positive or negative depending on the score. Given a novel enzyme-substrate pair, the model first detects all known substrates present in the training dataset which have Tanimoto similarity greater than a preselected substrate similarity threshold. Then it lists all enzymes present in the training set which are known to interact with the detected substrates. Finally, the model calculates pairwise similarity scores between the given enzyme and all the listed enzymes using Biopython’s pairwise module (5). If the similarity score between any of the listed enzyme and the given enzyme is higher than a preselected protein similarity threshold value, the enzyme-substrate pair is flagged as interacting, else it is predicted as a non-interacting/negative pair.

Note: The model is computationally quite intense. The pairwise enzyme sequence similarity calculation is the most time consuming step that uses a dynamic programming algorithm to calculate similarity between two sequences. A single pairwise comparison takes ~10.5 seconds to run. With 3408 unique enzymes in the training set and 29429 unique enzymes in the test set, constructing a pairwise similarity matrix will take 3408\*29429\*10.5 secs to run on a single core. Even if we parallelize over 24 cores, the time estimate to construct the matrix is ~507 days on a single node. Hence, we have restricted ourselves to evaluate the model on 1000 random interaction pairs both from validation and test dataset.

### Kernel Density Estimation

Kernel Density Estimation (KDE) is a non-parametric method which can be used to estimate the probability density function of a multivariate random variable. Given a sample of independent and identically distributed observations, , of a random variable, the estimated density function at a point is:

Here, is the kernel function, is the bandwidth, a smoothing parameter, and is the number of observations. KDE was trained on the positive enzyme-substrate interaction pairs present in our training set. The enzyme sequences were numerically encoded through the Dipeptide Deviation from Expected mean feature extraction technique and the substrates were encoded by their Morgan fingerprints. The numerical representations were concatenated into a single vector which represented a single observation. Positive interactions in the training set and all interactions in the validation and test sets were numerically encoded following which they were used to train and evaluate the KDE model. The kernel function and bandwidth hyperparameters were selected by observing model performance on validation dataset.

### DeepConv-DTI

We adopted DeepConv-DTI model’s architecture, developed by Lee et. al. (6) to predict drug-target protein interactions, for the purpose of predicting enzyme-substrate interactions. DeepConv-DTI employs convolutional neural network to numerically encode raw protein sequences into fixed length vectors. Similar to Lee et. al.’s work, we encoded the substrates by calculating their Morgan fingerprint. Finally, the encoded protein and substrate feature vectors were concatenated and passed as input to a feed forward neural network layer that is trained to predict interaction propensity of novel enzyme-substrate pairs. The optimized hyperparameters reported by Lee et. al. in the original paper was used to train the model.

# Results

Model performance was measured by its ability to correctly recover positive or negative interactions. The accuracy scores obtained by the three models on validation and test dataset is shown in Table 1.

Table : Accuracy scores achieved by the three models on validation and test set. Tanimoto model is evaluated only on 1000 random pairs both from the validation and test set due to computational constraints.

|  |  |  |  |
| --- | --- | --- | --- |
| Model \ Dataset | Validation (positive) | Validation (negative) | Test (positive) |
| Tanimoto | 0.8 | 0.67 | 0.68 |
| KDE | 0.5 | 0.57 | 0.36 |
| DeepConv-DTI | 0.56 | 0.6 | 0.46 |

The results indicate that Tanimoto model outperforms both KDE and DeepConv-DTI. However, it is evaluated only on ~16% interactions from the validation set and ~1.5% interactions from the test set due to computational constraints. Moreover, from the results, it seems that there might be multiple homologous interactions or interactions where the protein-substrate pairs are highly similar to other protein-substrate pairs. These interactions should be eliminated before assessing Machine Learning model performance as directed by multiple journals.

Since the Tanimoto model becomes computationally infeasible on a larger scale, it is essential to improve the performance of KDE and DeepConv-DTI such they are on par with or exceed Tanimoto similarity model’s performance. KDE model can be improved by substituting the DDE based protein feature extraction technique with a deep representation learning technique such as UniRep, proven to enhance performance on several protein engineering prediction tasks (7). DeepConv-DTI’s modeling architecture can be supplemented with a recurrent neural network layer after the convolutional neural network layer which can boost the model’s ability to glean further spatial and temporal relation from the protein sequences as illustrated in previous studies (8).

# References

1. Placzek S, Schomburg I, Chang A, Jeske L, Ulbrich M, Tillack J, et al. BRENDA in 2017: New perspectives and new tools in BRENDA. Nucleic Acids Res. 2017;45(D1).

2. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. Vol. 28, Nucleic Acids Research. 2000.

3. Landrum G. RDKit: Open-Source Cheminformatics Software. Http://Www.Rdkit.Org/. 2021.

4. Saravanan V, Gautham N. Harnessing computational biology for exact linear B-cell epitope prediction: A novel amino acid composition-based feature descriptor. Omi A J Integr Biol. 2015;

5. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11).

6. Lee I, Keum J, Nam H. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. PLoS Comput Biol. 2019;15(6).

7. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. Nat Methods. 2019;16(12).

8. Pfeiffenberger E, Bates PA. Predicting improved protein conformations with a temporal deep recurrent neural network. PLoS One. 2018;13(9).