ESCI upf.
School of International Studies

MEMÒRIA DEL TREBALL DE FI DE GRAU DEL GRAU (ESCI-UPF)

Prediction of ADMET properties with Machine Learning: a trustworthy and explainable approach

AUTOR/A: Zinnera Tariq	**NIA: 104632**
GRAU EN BIOINFORMÀTICA	
CURS ACADÈMIC: 2021-2022	
DATA: 21/06/2022	
TUTOR/S: Alexis Molina	

Prediction of ADMET Properties with Machine Learning: a trustworthy and explainable approach

Zinnera Tariq

Scientific director: Alexis Molina[1]

[1] IT Department, Address: Av. de Josep Tarradellas, 8-10, 3-2, 08029 Barcelona

Abstract

Motivation: Pharmaceutical research and development (R&D) is now a high-risk investment that is prone to unexpected, even disastrous, failures at various phases of drug development. Efficacy and safety issues linked to absorption, distribution, metabolism, excretion (ADME) qualities, and different toxicities (T), are significant causes of drug development failures. Therefore, conducting an ADMET analysis on time is critical. The ability to predict these features quickly and accurately allows researchers to rule out compounds that may present problems with ADMET and prioritize which compounds to produce and test. Given the current R&D model's tremendous complexity, molecular modeling methodologies to find patterns in ADMET data and convert them into knowledge have been pursued. In this project, we have generated and validated models to predict ADMET properties and determine the realm of applicability domain and explainability of such models.

Results: This paper reported a dataset of 8997 compounds for 21 ADMET points. For classification models, to differentiate between the two types of classes, 180 descriptor- and fingerprint-based models were developed. The statistical results showed that the model based on substructure key fingerprints, MACCS (Molecular ACCess System) keys outperformed the others, obtaining an overall Matthew's correlation coefficient (MCC) of 0.8716 and Area Under the Curve (AUC) of 0.9219 for the test set. In addition, nine regression models based on descriptors and fingerprints were generated. With an R^2 = 0.9146 for the test set, the MACCS-based model again outperformed the others. However, the applicability domain analysis showed that MACCS-based model prediction might be unreliable. Property-based models, on the other hand, were more reliable. Furthermore, 20 GNN-based classification and regression models were created. These models performed admirably, with an average ACC = 0.9297 for the classification models and R^2 = 0.8874 for the regression model's test set. Furthermore, the attribution score methods and the GNNExplainer were used to identify important structural fragments related to the Blood-Brain-Barrier (BBB) and the cytochrome enzyme, CYP2C9 inhibitor.

1 Introduction

Drug discovery and development is a time-consuming and costly process. With the development of *in silico* methods, the number of new chemical entities (NCEs) has increased in recent years. [1] However, many drug candidates still do not become drugs. This is mainly attributed to pharmacodynamic (PD) issues such as selectivity or efflux and pharmacokinetic (PK) problems such as poor metabolic properties and toxicity of drug candidates. [2]

Traditionally, a drug candidate's ADMET (absorption, distribution, metabolism, excretion, and toxicity) properties were measured after its potency against a specific target was determined. [3] Unfortunately, undesirable adverse effects were frequently detected at this stage, necessitating a new round of molecular design and syntheses or even the complete termination of the project. It was estimated that 40-60% of NCE failures are

due to poor ADMET profiles [4], emphasizing the importance of early ADMET evaluation in the drug development process. Currently, the potency and ADMET profiles of molecules are typically tested at the same stage, allowing undesirable compounds to be removed earlier in the drug discovery and development process. [5,6] A fine balance between drug candidates and their ADMET, profiling during the drug molecule's synthesis can help avoid late-stage drug failure in the drug discovery process. Thus, early detection of PK/PD properties, drug similarity, and ADMET analysis can save money and time while ensuring the safety and stability of the candidate drug. [7,8]

ADMET data is considered an essential component of new drug discovery and development. Both *in vitro* and *in vivo* models provide parameters related to drug ADMET properties, which can be used to predict drugs' behavior after administration. The idea about ADMET parameters for any compound is to have a significant impact before entering preclinical trials to reduce drug withdrawals from specific stages of pre-clinical and clinical trials. Therefore, most pharmaceutical industries rely heavily on earlier evaluation through *in silico* prediction tools, including regression and classification-based approaches, for machine learning (ML) and deep learning (DL) methods. [9] These computational prediction tools have provided enough information over the last two decades to demonstrate that well-established predictive models can predict ADMET profiles and drug similarities well before drug synthesis. [7,10]

Currently, there are several free and commercial computational tools for predicting ADMET properties. [11] However, these tools are limited in application and are not yet sufficiently reliable. [12] Among the popular tools, ADMETLab [13] offers 53 prediction models based on graph-structured data generated using a multi-task graph attention network. The method can generate customized fingerprints for a specific activity using general attributes. SwissADME [14], another web tool, evaluates the pharmacokinetics and drug-likeness of small molecules. The predictions are based on fragmental approaches and ML-based binary classification methods for additional ADMET properties. Drug discovery and environmental risk assessment models are constructed using MACCS (Molecular ACCess System) keys and Morgan fingerprints in ADMETSar. [15] The toxicity models employed in ProTox [16] are based on chemical similarities between compounds with known toxic effects and the presence of toxic fragments. Other models for hepatotoxicity, cytotoxicity, mutagenicity, and carcinogenicity rely on fingerprints (MACCS/Morgan). In the vNN server [17], extended-connectivity fingerprints predict 15 ADMET properties, with models trained using the variable nearest neighbor method. pkCSM [18], on the other hand, develops predictive models of central ADMET properties using graph-based signatures. Other software such as MDCKPred [19], CarcinoPred-EL [20], and CapsCarcino [21] focuses on a single property such as the prediction of permeability coefficient and carcinogenic compounds. The models' molecular representations include a variety of molecular and physicochemical descriptors such as fingerprints, graph signatures, and other 2D/3D indices. [22,23] Fingerprint representations, used as an alternative to descriptors in Quantitative structure-property relationships (QSPR) studies, have gained popularity due to their ease of computation and predictive value.

Historically, the majority of effort in developing *in silico* prediction tools has been focused on providing high accuracy, with a good model that can predict the correct value most of the time. Such a viewpoint is acceptable when performing a large number of predictions with a return on investment proportional to the number of correct outcomes, for instance, with virtual screening. However, a model's overall performance is ineffective when estimating the confidence in a specific individual prediction in contexts such as human safety assessment, regardless of how generalized it appears after validation. [24]

Model predictions on new compounds with descriptor values outside the training data's descriptor (feature) space may be unreliable. As a result, knowing the limit beyond which the model can reliably extrapolate is essential. [25] The Applicability Domain (AD) concept overcomes this issue by defining the domain in which the model can provide correct predictions. Predictions of compounds' biological activity outside this domain are rejected because they are likely incorrect. [26] Most proposals used to measure the AD are based on distance-based methods. Distance-based methods calculate the distance between a new compound and its k-nearest neighbors (or the centroid of the training set) using distance measures (e.g., Tanimoto or Euclidean). A distance-based threshold is used to determine whether or not the new compound is within the AD. Predictions of any compound above the threshold are regarded as unreliable. The downside of this method is that the threshold value is often arbitrary. [27]

Deep Learning (DL) is emerging as a critical technology for performing various tasks in cheminformatics. [28-30] With the recent development of artificial intelligence (AI) and DL, the application of DL approaches for various predictions such as virtual screening, [31] quantitative structure-activity relationship (QSAR) studies, [32] and ADMET prediction has been practically demonstrated. [33]

The models developed by DL regarding ADMET prediction can be roughly classified into two categories: descriptor-based and graph-based. [34] In the case of descriptor-based DL models, molecular descriptors or fingerprints similar to those used in traditional QSAR models are used as input. Then a specific DL architecture is used to train a model. [32] In the case of graph-based DL models, basic chemical info encoded by molecular graphs is used as input, and a graph-based DL algorithm, such as graph neural networks (GNNs), is then used to train a model. GNNs use a graph-structured representation of the original molecule as input data, with atoms as nodes and bonds as graph edges. [35] The key feature of GNN is its ability to learn task-specific representations automatically using graph convolutions without using traditional handcrafted descriptors or fingerprints.

GNNs have been shown to perform well in predictions based on molecular structures [35,36], and numerous studies have shown that GNNs can outperform traditional descriptor-based methods. [34,37-42] In particular, graph convolutional networks (GCNs), a type of GNN, performed admirably in various applications. [43]

Despite their promise, GNNs remain of limited acceptance in drug discovery partly due to their lack of interpretability. [44] The development of explainable artificial intelligence (XAI) techniques has overcome this limitation. The goal of XAI techniques is to help understand how the model arrived at a particular answer and why the answer provided by the model is acceptable. [45] XAI methods could aid in developing GNNs in drug discovery applications, particularly for property prediction tasks, by quantifying the molecular substructures that are critical for a given prediction and explaining how reliable a prediction is. [44]

An example of the XAI technique is Integrated Gradients (IG). [44] IG aims to explain the relationship between a model's predictions and features. It has numerous applications, such as understanding feature importance, detecting data skew, and debugging model performance. Because of its broad applicability to any differentiable model, ease of implementation, theoretical justifications, and computational efficiency relative to alternative approaches, IG has become a popular interpretability technique. Another common gradient explainer is saliency maps. This explanation highlights the most reactive features and is likely to change the output quickly.

In this project, the data set of 8997 compounds for 21 ADMET endpoints from a public database was used for modeling using classic ML methods, including a DL method: GNN. In addition, the model performance was validated by the internal and external validations. Afterward, the applicability domain was determined for the models' reliable application in predicting new chemicals. Finally, the important structural fragments related to the blood-brain barrier (BBB) and the CYP2C9 inhibitor were recognized by GNNExplainer [46] and by the following attributions methods: IG and saliency maps.

1.1 Objectives

The primary goal of the project is to develop and implement a machine learning algorithm for predicting ADMET properties, as well as to define the predictive model's applicability domain and provide an explanation of the model's explainability.

2 Methods

2.1 Data Collection and Preparation

The development of a successful ADMET prediction model requires an accurate and relevant dataset suitable for the model. In general, there exists a relatively small number of ADMET data, especially public datasets, with a desirable quality of data, diversity of the investigated structures, and which are large enough to permit sufficient validation of the derived model. To obtain as much data as possible for model training, a comprehensive data retrieval was conducted by crossing the following database: ChEMBL [47], BindingDB [48], Comptox [49], IMPPAT [50], PubChem [51], and DrugBank. [52] The molecules from the mentioned databases were integrated into a single database, and the duplicated molecules were removed.

2.2 Descriptors Calculation

Molecular descriptors and fingerprints play a crucial role in the development of successful prediction models. The accumulated experience in bioinformatics studies demonstrates that ML predictions rely heavily on effective

molecular representations. [53] Thus, molecular descriptors and fingerprints were computed to further model building. RDKit [54] – an open-source cheminformatics software - and Scopy [55] – a python package - were used to calculate the 2D descriptors (physicochemical properties) for the molecules.

Molecular fingerprints are a particularly complex type of descriptor, with each bit representing a feature's presence (1) or absence (0), alone or in conjunction with other bits in the bit string. [56] There are several types of molecular fingerprints depending on how the molecular representation is transformed into a bit string. Most methods use only the 2D molecular graph and are thus referred to as 2D fingerprints; however, some techniques can also store 3D information. The three main approaches are structure keys-based fingerprints, circular fingerprints, and topological or path-based fingerprints. We will focus on the first two in this study.

Substructure key-based fingerprints set the bits of the bit string based on the presence of specific substructures or features from a given list of structural keys in the compound. This usually means that these fingerprints are most valuable when applied to molecules that are likely to be primarily covered by the given structural keys, but not so much when applied to molecules that are unlikely to contain the structural keys because their features would not be represented. The number of structural keys determines their number of bits, and each bit relates to the presence or absence of a single given feature in the molecule. KRFP (Klekota-Roth Fingerprint) and MACCS keys are fingerprints based on substructure keys used in this project. MACCS keys are 166-bit 2D structure fingerprints commonly used to measure molecular similarity. Moreover, KRFP is 4086-bit fingerprints.

Circular fingerprints are created by exhaustively enumerating all circular fragments grown radially from each heavy atom of the molecule up to the specified radius and hashing these fragments into a fixed-length bit-vector. In this project, two types of circular fingerprints were used; ECFP (Extended-Connectivity Fingerprint) and FCFP (Functional-Class Fingerprint), with diameters of 2, 4, and 6. ECFP is based on the Morgan algorithm [57] and was developed specifically for use in structure-activity modeling. It represents circular atom neighborhoods and generates variable-length fingerprints. It is most commonly used with a diameter of 4 inches (ECFP4). Although some benchmarks have shown minor performance differences between the two [58], a diameter of 6 (ECFP6) is also commonly used. FCFP is a variant of ECFP that indexes the role of an atom in the environment rather than the atom itself. Hence, the fingerprint cannot distinguish between atoms or groups that perform similar functions. This enables them to be used as pharmacophoric fingerprints.

RDKit and PyFingerprint [59] were used to calculate the following fingerprints: MACCS keys, ECFP2, ECFP4, ECFP6, FCFP2, FCFP4, FCFP6, and KRFP.

2.3 Model Development and Validation

This study implemented a total of 189 prediction models, including 180 classification and nine regression models. Based on the different sets of fingerprints and descriptors mentioned earlier, nine models were trained for each AMDET endpoint.

For modeling, PyCaret was used. Pycaret is an open-source, low-code Python library aiming to automate ML model development. The library contains over 70 automated open-source algorithms and over 25 pre-processing techniques that can help build high-performing ML models. It supports supervised learning (classification and regression), clustering, anomaly detection, and natural language processing.

Validation of *in silico* models is a crucial step in understanding models' reliability when making predictions for new molecules that are not present in the training data set. Model validation could be either internal (using the training set) or external (using separate unseen data). Internal validation methods include cross-validation. [60] The external method requires evaluating model performance on a separate test dataset using statistical assessments. Thus, the dataset was divided into training and test sets for this purpose, with 90% of compounds serving as the training set and 10% serving as the external validation test set. The same training and test set were used for all prediction models. Both data sets for each property are evenly distributed, as shown by the histograms (*Supplementary File 1*).

PyCaret's workflow is demonstrated in **Figure 1**. Data must be cleaned and formatted before being used in ML models; raw data cannot be used directly. Nevertheless, we do not have to clean the data manually because PyCaret does it for us. Therefore, PyCaret's first step (*setup()*) is in charge of initializing the environment and preprocessing the data. It also takes care of the internal

validation by performing cross-validation (10-fold by default). The train-test split for internal validation is also specified in this step. PyCaret's default split is 70:30. The default ratio was used in this project. As a result, the training set contained 5667 compounds, and the test set included 2430.

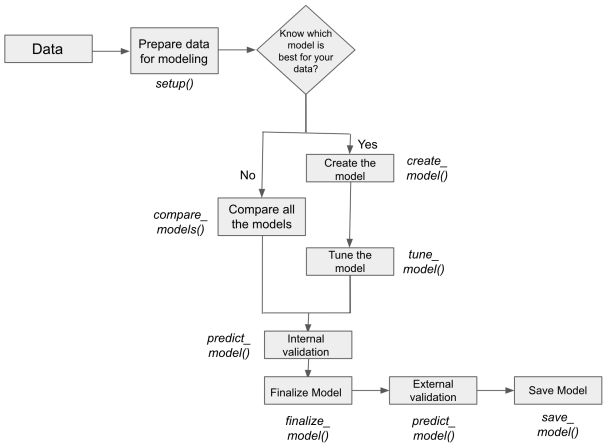

Figure 1. ***Workflow Diagram of PyCaret.*** *This diagram summarizes the main step of PyCaret.*

Once the setup is complete, the recommended starting point for modeling is to compare all models to evaluate performance (unless you know exactly what model you need, which is not the case here)—this function trains and scores all models in the model library using stratified ten-fold cross-validation for metric evaluation. As a result, a table is generated with the average Accuracy (ACC), the area under the ROC curve (AUC), Recall, Precision, F1, Kappa, and Matthews correlation coefficient (MCC) for the classification models, and R-square (R^2), mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), root mean squared log error (RMSLE), and mean absolute percentage error (MAPE) for both classification and regression models, across the 10-folds. Furthermore, the best-performing model is highlighted. The difference between test and cross-validation was checked to ensure the selected model was not overfitted. A model is considered overfitted if there is a significant difference between the test and cross-validation.

Following that, we can finalize the model; this function applies the model to the entire dataset used for the training, including the test sample (30 percent in this case). Finally, we can predict the finalized model for external validation on previously unseen data (the 10% we separated at the start and did not include in model training).

The test set's AUC, MCC (Eq. 1), overall ACC (Eq. 2), and 10-fold cross-validation of the entire dataset were used to evaluate all classification models. AUC is a single scalar value that measures the overall performance of a binary classifier. MCC is a model quality measure that returns a value between -1 and 1. MCC value 0 represents the average or random prediction, -1 represents the worst prediction, and +1 represents perfect prediction. ACC denotes the proportion of correct predictive positive and negative classes. However, this only applies to models trained on datasets with relatively balanced samples across classes.

$$MCC = \frac{(TP\ x\ TN) - (FP\ x\ FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

$$ACC = \frac{(TP+TN)}{TP+FP+FN+TN} \quad (2)$$

where, TP: true positive, TN: true negative, FP: false positive, FN: false negative.

The regression models, on the other hand, were evaluated using R^2 (Eq. 3), RMSE (Eq. 4), and MAE (Eq. 5). The MAE gives simple information about the average magnitude of errors that can be expected from a model. However, because all errors are weighted equally, differences in error magnitudes are averaged out; thus, the MAE alone does not provide insight into the uniformity or variability of prediction errors. Metrics based on squared errors, such as the RMSE, magnify more significant errors and thus are more sensitive to outliers. When MAE and RMSE are considered together, they can provide information on the homogeneity or heterogeneity of errors: if the MAE and RMSE values are similar, this indicates prediction errors of relatively consistent magnitude; if the RMSE is significantly larger than the MAE, this indicates large fluctuations in the magnitudes of the errors.

$$R^2 = 1 - \frac{\Sigma(\widehat{y_i} - y_i)^2}{\Sigma(y_i - \bar{y})^2} \quad (3)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2} \quad (4)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i}) \quad (5)$$

where $\widehat{y_i}$ and y_i are the predicted and experimental values of the ith sample in the dataset; $\bar{y}$ is the mean value of all the experimental values in the training set.

2.4 Applicability Domain Evaluation

The applicability domain (AD) is another way to validate the model. The AD evaluation ensures that the model can reasonably and accurately predict certain compounds.[61]

An ideal AD approach estimates interpolation regions in multivariate space. Principal components analysis (PCA) can be used to develop multivariate models. PCA aims to reduce dimensionality and noise in large amounts of data while extracting important features. Additionally, it can be used as an outlier detection method. The Distance to model X (DModX) [62] and Hotelling's T2 statistics, in particular, are useful in detecting outliers.

DmodX with PCA and Logistic PCA was used in this project to investigate the presence of outliers in the dataset. DmodX, also known as residual standard deviation, indicates the distance between data in variable X space and the principal component model, measures data changes outside the model, and represents changes in samples that the model does not explain.

Additionally, DBSCAN [63] was also used to define AD. The DBSCAN algorithm is a well-known density-based data clustering algorithm. To cluster data points, this algorithm divides the data into high-density and low-density areas. Unlike other clustering algorithms, we do not need to provide the number of clusters required in advance with this algorithm. The DBSCAN algorithm groups the points based on distance measurement. An essential property of this algorithm is that it helps us track the outliers as the points in low-density regions; thus, it is not sensitive to outliers.

Before we can apply the DBSCAN model, we should first reduce the dimensionality of our data. In order to do that, PaCMAP (Pairwise Controlled Manifold Approximation) [64] was used. This dimensionality reduction method was chosen because it preserves the data's local and global structure in the original space.

Next, we need to obtain the following parameters: epsilon (Eps) and MinPoints. An epsilon value is the shortest distance between two points to be considered neighbors. To compute Eps, the Nearest Neighbours function was used to calculate the distance between each data point and its nearest neighbor. After that, the distances were sorted and plotted (*Supplementary Figure 1*). Initially, the highest value in the plot, 0.7, was used as the Eps value. Furthermore, we wanted to use DBSCAN for different Eps values. As a result, two values less than the Eps value (0,5 and 0.6) and two values greater than the Eps value (0,8 and 0.9) were used. However, the same outliers were obtained using the mentioned Eps values. After further investigation, we noticed that the number of outliers does not change between epsilon values of 0 and 2. So we end up using DBSCAN for Eps values of 2, 2.5, 3, 3.5, 4, 4.5, and 5. Moreover, MinPoints is the smallest number of points required to build a cluster. A cluster is only recognized if the total number of points exceeds or equals the MinPoints.

Following the completion of the DBSCAN clustering, we have three data points: a *core* point for which both parameters are fully defined, i.e., a point with at least Minpoints within the Eps distance from itself, and any data point that is not a core point but has at least one core point within Eps distance from it is considered a *border* point. The last type of data point is the *noise* point, defined as a point with less than Minpoints within Eps of itself.

DBSCAN clustering algorithmic steps are illustrated in **Figure 2**. The algorithms initiate by randomly selecting a point (x) from the data set and finding all the neighbor points within Eps from it. We consider x a core point if the number of Eps-neighbors is greater than or equal to MinPoints. x then forms the first cluster with its Eps neighbors. After forming the first cluster, we examine its points to determine their Eps -neighbors. If a point has at least MinPoints Eps-neighbors, we expand the initial cluster by incorporating those Eps-neighbors. This process is repeated until there are no more points to add to this cluster. This procedure is repeated until all core points have been assigned to a cluster. Finally, it iterates through all unattended points in the dataset, assigning them to the cluster nearest to them at Eps distance. A point is considered a noise point if it does not fit into any available clusters.

This alternative methodology for computing outliers and thus, providing insight into the AD for a given model aims to treat better highly dimensional binary data such as fingerprints and serve as a tailored AD evaluation tool through the epsilon parameter.

After calculating the outliers, the model was trained using PyCaret, with the outliers serving as the test set and the remaining data serving as the training set. The goal was to see how the model performed on a test set outside AD.

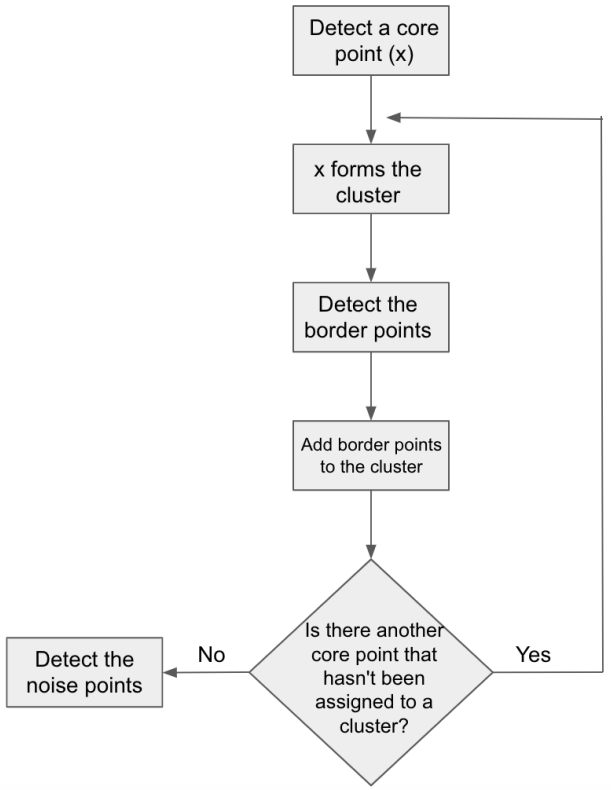

Figure 2. ***Workflow Diagram of DBSCAN.*** *This diagram summarizes the main step performed by the DBSCAN algorithm.*

Another method for validating the model is to use an external set and compare the similarity between our dataset and the external set using similarity metrics such as the Tanimoto index T (also known as the Jaccard coefficient) (Eq. 6). The Tanimoto index compares two compounds based on the number of common molecular fragments. Tanimoto similarity is determined by counting all unique fragments of a given length in two compounds. Tanimoto similarity between compounds A and B is defined as follows):

$$T(A,B) = \frac{\sum_{N}^{i=1}(x_{A,i} * x_{B,i})}{\sum_{N}^{i=1}(x_{A,i} * x_{A,i}) + \sum_{N}^{i=1}(x_{B,i} * x_{B,i}) - \sum_{N}^{i=1}(x_{A,i} * x_{B,i})} \quad (6)$$

where N is the number of unique fragments in both compounds, $x_{A,i}$ *and* $x_{B,i}$ *are the counts of the i-th fragments in the compounds A and B.*

It is expected that the molecules with a higher Tanimoto index will be inside the AD. We used an external set of 76145 compounds from ChEMBL for further analysis, selecting the 10% most similar to our data and the 10% most dissimilar, as determined by the Tanimoto index. Then PCA was performed on both sets, and DmodX was used to detect outliers. Finally, DBSCAN was performed with the following epsilon values: 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5.

2.5 Graph Neural Networks (GNNs)

GNN is a unique neural network architecture with the same basic principles as convolutional neural networks. However, it is primarily used to process and learn irregular and unstructured graph data. [65] GNNs aim to learn the representation of each node in the graph and then extract features of the nodes or graphs hierarchically before using the final features for application modeling by a sub-model, such as a multi-layer perceptron (MLP). GNNs use the graph structure to iteratively update the node representation from the node neighborhoods in a convolutional or equivalent fashion to obtain the final feature representation of the nodes or the graph. [65] Multiple graphical convolution (or equivalent) layers are typically stacked together to update the node representation to explore the deeper and more extensive information of the node's receiving domain.

Before the data were inputted into the GNN model, each molecule was transformed into an undirected graph $G(V, E)$, where $V = \{x1, x2, \ldots, xn\}$ is the node-set representing atoms, and E is the edge set representing chemical bonds. *Supplementary Table 1* lists the RDKit-generated atom and bond descriptors used as input node and edge features.

The data for each ADMET endpoint was randomly split to separate the training, internal, and external validation sets by an 8:1:1 ratio. For modeling, the PyTorch Geometric [66] library was used. PyTorch Geometric is an extension library to the popular DL framework Pytorch [67] and consists of various methods and utilities to ease the implementation of GNNs. The model was defined using a five-layer GCN, with each GCN layer enhanced by the activation function, Rectified Linear Unit (ReLU). Following that, two linear transformations were applied as a classifier to map the nodes to one of the two classes. After the first linear layer, a dropout layer with p=0.5 was used. The model was trained for a maximum of 500 epochs (training iterations) with optimizer Adam [68] at a learning rate of 0.001 and early stopping at a window size of 20, i.e., the training was terminated if the validation loss did not decrease for 20 consecutive epochs. Moreover, 10-fold cross-validation was applied to tune parameters.

The accuracy and the negative log-likelihood were used to evaluate the classification models. The regression model, on the other hand, was evaluated using R^2 (Eq. 3) and MSE. To further validate the models, we obtained experimentally validated data for the ADMET properties. The idea is to use this data to train GNN models; if they perform well, our models are reliable. Because it is difficult to obtain experimentally validated data for each property, we only obtained data to validate the following models: BBB, Caco-2, Carcinogenicity, Ames Test, human intestinal absorption (HIA), and hERG inhibitor. If the models predict these properties well, we can assume they will also predict the other properties well.

2.6 Explicability of the GNN Model

An attribution method, given a trained model and an input, assigns scores to each input feature that reflects the feature's contribution to the model prediction. Attribution scores reveal which features, in this case, atoms and atom pairs, were most important to the model's decision. In this study, the attribution scores were calculated and represented using Integrated Gradients (IG) [69] and Saliency. Consider the function $F: R^n \rightarrow [0, 1]$ of a deep neural network to demonstrate how IG works briefly. Given an input feature *x* (in our case, x is a molecule that has been divided into atoms and atoms pairs) and some baseline feature *x'*, the IG of *x* along the i-th dimension of *x* was defined as follows (Eq. 7):

$$a_i = (x_i - x_i`) \, x \int_{i=0}^{1} \frac{\partial F(x)}{\partial x_i} \Big|_{x = x' + tx(x - x')} dt \qquad (7)$$

where $\frac{\partial F(x)}{\partial x_i}$ *is the gradient of F along the* i*-th dimension of x.*

Furthermore, the Saliency computes attributions by taking the absolute value of the partial derivative of the target output with respect to the input. Additionally, GNNExplainer [70] was used to provide interpretable explanations for the predictions of the GNN-based model. GNNExplainer recognizes a compact subgraph structure and a small subset of node features that play an important role in GNN prediction.

2.7 Code and Data Availability

The data and the scripts used for this project are available on GitHub.

3 Results and Discussion

3.1 Data Collection

The assembled database contains 51,352,643. compounds with canonical SMILES, InChIKey, 37 experimental physicochemical properties, and 21 ADMET properties. The physicochemical and ADMET properties are listed in *Supplementary Tables 2* and *3*, respectively.

Once the database was assembled, we retrieved compounds for which we had available any experimental ADMET data. We obtained 8997 compounds with explicitly annotated ADMET data. These compounds come from DrugBank. [52]

3.2 Descriptors Calculation

It was not possible to calculate all the physicochemical properties (listed in *Supplementary Table 2* from the python packages mentioned in **Methods** section **2.2**. Hence, only 14 physicochemical properties (listed in *Supplementary Table 4*) were considered for further analysis as we could calculate them for each molecule.

3.3 Model Development and Validation

As stated in **Methods** section **2.3**, nine models were created for each ADMET property. One of those models was obtained for 2D descriptors, while the others were created using different molecular fingerprints calculated.

To ensure that the predictive model had good generalization, a test set, ten-fold cross-validation, and an external validation set were used for model validation. The best and worst-performing models for each property will be discussed in this section. The complete performance summary of the all the models can be found in *Supplementary Table 5*.

Table 1 summarizes the relevant performance metrics associated with the best-performing classification models. Light Gradient Boosting Machine (LGBM) was the best ML approach for most classification models. The remaining models performed best with Random Forest (RF) and Logistic Regression (LR).

All the classification models performed best with MACCS key fingerprints. All classification models achieved an AUC of 0.85 or higher, except for the Ames Test. Moreover, these models produced acceptable prediction accuracy, with 19 models achieving ACC values greater than 0.93. The average MCC value of these models is

Endpoint	ML Approach	Feature	AUC	ACC	MCC
Ames Test	LGBM	MACCS	0.8146	0.9367	0.6984
Biodegradation	LGBM	MACCS	0.9425	0.9611	0.8889
Blood Brain Barrier	LGBM	MACCS	0.9527	0.9644	0.9055
Caco-2 Permeable	LGBM	MACCS	0.952	0.9533	0.904
Carcinogenicity	RF	MACCS	0.9661	0.9956	0.9632
CYP450 1A2 inhibitor	LGBM	MACCS	0.9265	0.9489	0.8674
CYP450 2C19 inhibitor	LGBM	MACCS	0.9056	0.9567	0.8333
CYP450 2C9 inhibitor	LGBM	MACCS	0.9077	0.9622	0.8334
CYP450 2D6 inhibitor	LGBM	MACCS	0.9046	0.9833	0.8338
CYP450 2D6 substrate	LGBM	MACCS	0.8744	0.9922	0.8398
CYP450 3A4 inhibitor	RF	MACCS	0.8958	0.9678	0.8265
CYP450 3A4 substrate	LGBM	MACCS	0.9768	0.9767	0.9533
CYP450 inhibitor promsicuity	LGBM	MACCS	0.9137	0.9589	0.8591
hERG inhibitor (predictor I)	LR	MACCS	0.8571	0.9978	0.8442
hERG inhibitor (predictor II)	LGBM	MACCS	0.9511	0.9856	0.9236
Human Intestinal Absorption	LR	MACCS	0.9202	0.9733	0.8489
P-glycoprotein inhibitor I	LGBM	MACCS	0.943	0.96	0.8978
P-glycoprotein inhibitor II	LGBM	MACCS	0.9345	0.96	0.882
P-glycoprotein substrate	LR	MACCS	0.9774	0.9811	0.9573

Table 1. *Performance metrics for the best-performing classification models.*

0.87, and 57% of models have an MCC greater than 0.85. Generally, these models can predict the ADMET-related properties of molecules pretty accurately. However, we can see that the model for CYP2C9 substrate did not perform as expected (see *Supplementary Table 5*), most likely due to the unbalanced data. As shown in *Supplementary Table 1*, the data for this cytochrome enzyme includes 8994 non-substrates and only three substrates. Hence, this property was discarded for further analysis. The regression model also achieved the best performance with MACCS key fingerprints.

The model achieved an R^2 of 0.9146, an MAE of 0.1128, and an RMSE of 0.187 with Random Forest (RF). The external validation, internal validation, and cross-validation demonstrated that the constructed models could accurately predict the property values to some extent.

Table 2 summarizes the worst-performing models and performance measures for the classification models. We can observe that the worst-performing model for each property was one of the circular fingerprints (ECFP or FCFP). Nonetheless, these models still give good predictions. For classification models, forty-four percent of the models had an AUC of 0.85 or higher. Furthermore, 18 models achieved ACC values greater than 0.89. The average MCC value of these models is 0.72, with 44% having an MCC greater than 0.75.

However, we can see that the model for hERG inhibitor I did not perform as expected with any of the remaining fingerprints (*see Supplementary Table 5*). In addition, the MCC (0.3767) and AUC (0.5714) values for the descriptor-based models were low, most likely due to unbalanced data. The training data for this toxicity property includes 8032 weak and only 56 strong inhibitors, as shown in *Supplementary Table 1*. Furthermore, the test set contains 893 weak and only seven strong inhibitors. Although we obtained good statistical results using MACCS keys, it is most likely overfitted. Thus, we cannot rely on the predictive ability of this model. In addition, the regression model performed the worst with a circular fingerprint-based model, ECFP6, with an R^2 of 0.7555 and 0.3164 and 0.2391 MSE and RMSE, respectively.

In general, all obtained models correctly predicted most of the test set properties, with an overall prediction accuracy of more than 68 percent (for the classification models) and an R2 greater than 0.75 (for the regression model). However, MACCS key models outperformed the others, while circular fingerprints performed the worst.

Endpoint	Feature	AUC	ACC	MCC
Ames Test	ECFP4	0.6838	0.8989	0.4743
Biodegradation	ECFP4	0.8823	0.9178	0.7646
Blood Brain Barrier	FCFP4	0.8671	0.9111	0.7579
Caco-2 Permeable	FCFP6	0.9086	0.9133	0.8212
Carcinogenicity	FCFP6	0.9444	0.9844	0.8751
CYP450 1A2 inhibitor	ECFP6	0.8784	0.9156	0.7789
CYP450 2C19 inhibitor	ECFP6	0.8226	0.9278	0.7099
CYP450 2C9 inhibitor	FCFP6	0.8083	0.9367	0.7022
CYP450 2D6 inhibitor	ECFP4	0.7696	0.9644	0.6163
CYP450 2D6 substrate	ECFP6	0.7066	0.9811	0.558
CYP450 3A4 inhibitor	ECFP6	0.8051	0.9433	0.6805
CYP450 3A4 substrate	FCFP4	0.9456	0.94363	0.8917
CYP450 inhibitor promsicuity	ECFP6	0.8301	0.9144	0.7008
hERG inhibitor (predictor II)	FCFP4	0.8474	0.9622	0.7892
Human Intestinal Absorption	ECFP4	0.8016	0.9489	0.686
P-glycoprotein inhibitor I	ECFP6	0.8573	0.8933	0.7263
P-glycoprotein inhibitor II	ECFP6	0.8149	0.8922	0.6711
P-glycoprotein substrate	ECFP2	0.9256	0.9333.	0.8498

Table 2. *Performance metrics for the worst-performing classification models.*

On the other hand, unbalanced data was the cause of the models' failure to make good predictions. Therefore, we can conclude that the accuracy of ADMET profiling prediction is determined by the datasets and modeling tools used to create the models. Thus, the dataset used for *in silico* ADMET prediction should be highly diverse and large enough to be considered a global dataset.

3.4 Applicability Domain Evaluation

Different approaches were applied to define the AD. To begin, PCA was performed not only on the MACCS keys fingerprints data but also on the rest of the fingerprints and molecular descriptors data to determine why the model performed best with the MACCS keys fingerprints and poorly with the other fingerprints or descriptors. Then, DmodX was used to detect the outliers.

As outlined in the scores plot of data using MACCS keys fingerprints (**Figure 3**), both train and test data sets are evenly distributed in the chemical space. Furthermore, the data using physicochemical properties (*Supplementary Figure 2*) and KRFP (Supplementary Figure 3) are equally distributed. On the other hand, the data using circular fingerprints (*Supplementary Figures 4-9*) are not evenly distributed. This could be one of the reasons we had poor results with the circular fingerprint-based models.

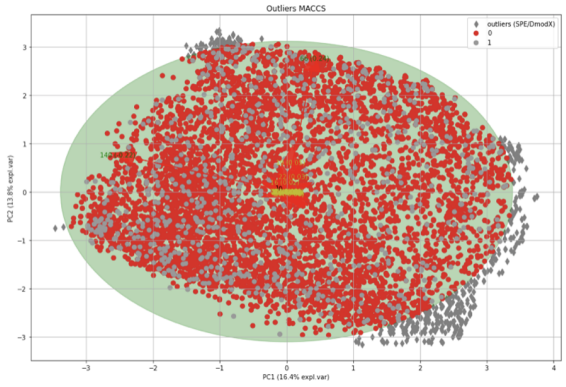

Figure 3. ***Scoring plot of the first two principal components in the training and test set for MACCS keys fingerprints data.*** *The training set is represented by the red circle points, while the gray circle points represent the test set. Outliers are shown as gray rhombus points.*

Table 3 summarizes the distribution of the training and test data inside and outside the AD determined by DmodX. Six hundred forty-four compounds are outside the AD for the data fingerprints using MACCS keys; more than 900 compounds are outside the AD for the remaining fingerprints. However, regarding AD coverage, MACCS keys have the lowest coverage (34%) for the test, while the rest have AD coverage above 84%. The descriptor-based data has achieved the highest AD

coverage (96.80%) for the test set. This suggests that the predictions for the best-performing model in the test set may not be completely reliable.

Dataeset	Inside AD		Outside AD		AD coverage(%)	
	Train	Test	Train	Test	Train	Test
Property-based	7785	871	312	29	96.15	96.80
MACCS	8047	306	50	594	99.39	34
ECFP2	7050	777	1047	123	87.07	86.34
ECFP4	7020	777	1077	123	86.70	86.34
ECFP6	7064	782	1033	118	87.24	86.89
FCFP2	7195	800	902	100	88.86	88.89
FCFP4	7198	800	899	100	88.90	88.89
FCFP6	7234	803	863	97	89.34	89.22
KRFP	7002	757	1095	143	86.48	84.11

Table 3. *The number of compounds inside and outside the AD determined by DmodX in the training and test sets.*

Following that, models were trained using MACCS outliers as the test set (644 compounds) and the remaining data as the train test (8353 compounds) to further investigate the best-performing model reliability. The models were trained using the best-performing models from the previous section (**Table 1**). *Supplementary Table 6* summarizes the classification and regression models' statistical results. In addition to hERG inhibitor I and CYP2C9 substrate, four more cytochrome enzymes failed to predict the model. This was due to the unbalanced data, as we used less than 8% of the data as the test set, and the data split was not done using a train-test split method, so the data was not well distributed. When compared to the best-performing models, the remaining classification models performed poorly. The average MCC of these models fell from 0.86 to 0.65, while the average AUC decreased from 0.92 to 0.8. The average ACC was not significantly affected, dropping from 0.96 to 0.94. For the regression model, the R^2 was dropped from 0.9146 to 0.6236. This suggests that we can trust the AD space defined by this method.

Secondly, logistic PCA was applied to the MACCS data because it is a useful tool for exploring relationships within a multivariate binary data set. DmodX was used once more to detect outliers. Outside the AD defined are 1335 outliers: 1230 from the training set and 135 from the test set. The AD coverage for the training and test sets is 84,81 and 85%, respectively. Then, for each endpoint, models were trained using the outliers as the test set and the remaining data as the training set (7662 compounds). *Supplementary Table 7* summarizes the classification and regression models' statistical results. Compared with the best-performing model, the ACC, AUC, and MCC have increased for the classification model. Nevertheless, the R^2 for the regression model has decreased significantly, from 0.9146 to 0.7470. The performance of these models was expected to be poor, but this did not occur. Thus, we can conclude that this method is untrustworthy for defining the AD.

The DBSCAN algorithm was then applied to MACCS data with different epsilon values. **Table 4** shows that the Eps value increases so does the AD coverage for the train and test sets. Furthermore, the DBSCAN outliers were used as the test sets for modeling. *Supplementary Table 8* summarizes the classification and regression models' statistical results.

For *eps=2,* 80 percent of the classification models performed worse than the best-performing models. The average MCC of these models fell from 0.87 to 0.80, while the average AUC decreased from 0.96 to 0.88. The average ACC was not significantly affected, dropping from 0.97 to 0.96. For the regression model, the R^2 was dropped from 0.9146 to 0.741. For *eps=2.5,* 63 percent of the classification models performed worse than the best-performing models. The average MCC of these models fell from 0.86 to 0.78, while the average AUC decreased from 0.91 to 0.87. The average ACC was not significantly affected, dropping from 0.97 to 0.96. For the regression model, the R^2 was dropped from 0.9146 to 0.7425.

Epsilon value	Inside AD		Outside AD		AD coverage(%)	
	Train	Test	Train	Test	Train	Test
$eps = 2$	7197	0	900	900	88.18	0
$eps = 2.5$	7297	108	800	792	90.12	12
$eps = 3$	7403	197	1397	703	91.43	21.89
$eps = 3.5$	7461	267	636	633	92.15	29.67
$eps = 4$	7573	374	524	526	93.53	41.56
$eps = 4.5$	7694	484	407	416	94.97	53.78
$eps = 5$	7778	578	319	322	96.06	64.22

Table 4. *The number of compounds inside and outside the AD determined by DBSCAN for the different values of Eps.*

For *eps=3,* 68 percent of the classification models performed worse than the best-performing models. The average MCC of these models fell from 0.86 to 0.75, while the average AUC decreased from 0.91 to 0.86. Moreover, the average ACC decreased from 0.97 to 0.88. For the regression model, the R^2 was dropped from 0.9146 to 0.8632. For *eps=3.5,* 63 percent of the classification

models performed worse than the best-performing models. The average MCC of these models fell from 0.85 to 0.76, while the average AUC decreased from 0.91 to 0.86. Moreover, the average ACC decreased from 0.97 to 0.88. For the regression model, the R^2 was dropped from 0.9146 to 0.8443.

For *eps=4, 78* percent of the classification models performed worse than the best-performing models. The average MCC of these models fell from 0.87 to 0.79, while the average AUC decreased from 0.92 to 0.88. Furthermore, the average ACC decreased from 0.97 to 0.93. For the regression model, the R^2 was dropped from 0.9146 to 0.8309. For *eps=4.5,* 61 percent of the classification models performed worse than the best-performing models. The average MCC of these models fell from 0.88 to 0.8, while the average AUC decreased from 0.92 to 0.88. The average ACC was not significantly affected, dropping from 0.974 to 0.966. For the regression model, the R^2 was dropped from 0.9146 to 0.7869.

For *eps=5, 67* percent of the classification models performed worse than the best-performing models. The average MCC of these models fell from 0.88 to 0.78, while the average AUC and ACC did not change significantly, dropping from 0.874 to 0.869 and from 0.974 to 0.967, respectively. For the regression model, the R^2 was dropped from 0.9146 to 0.5186.

To summarize, most classification models performed poorly with different epsilon values, with a significant difference in AUC and MCC compared to the best-performing models. Furthermore, as the value of epsilon increased, so did the AD coverage for the test set. This makes sense because the higher the epsilon, the more outlier data points were included in the training set, increasing the AD for this particular set. This implies that, depending on the value of epsilon.

To further determine how reliable these AD techniques (DmodX with PCA and DBSCAN) are, they were tested on two external sets: one similar to our data and one that was not. **Figures 4** and **5** show the score plot for the similar and the dissimilar set, respectively.

The data for a similar set is evenly distributed. However, the external set is mainly on the right side of the score plot for the dissimilar set. Although the score plots appear to show that the dissimilar set has more compounds outside the AD than the similar set, both sets yielded a similar number of compounds outside the AD, 1083 for the similar set and 956 for the dissimilar set. However, the similar set had an AD coverage of 81.76 % (165 compounds), while the dissimilar set had a coverage of 37.44 % (563 compounds). Hence, we can rely on the domain of applicability given by DmodX as more compounds from a similar set are inside the AD, and more compounds from a dissimilar set are outside the AD.

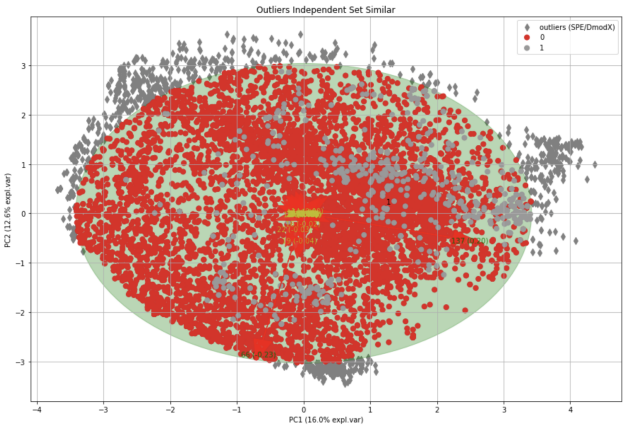

Figure 4. ***Scoring plot of the first two principal components in the dataset and the similar set.*** *The dataset is represented by the red circle points, while the gray circle points represent the compounds of the similar set. Outliers are shown as gray rhombus points.*

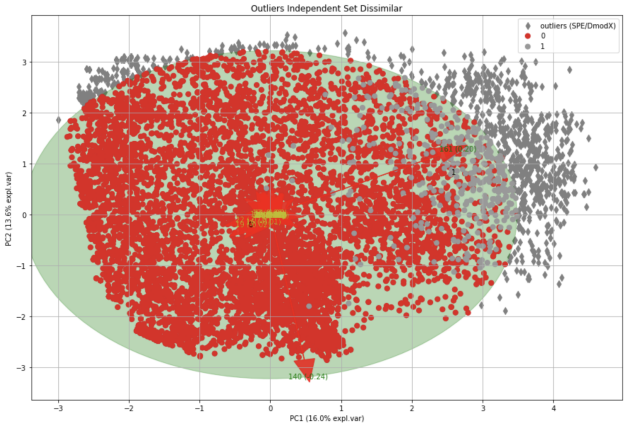

Figure 5. ***Scoring plot of the first two principal components in the dataset and the dissimilar set.*** *The dataset is represented by the red circle points, while the gray circle points represent the compounds of the dissimilar set. Outliers are shown as gray rhombus points.*

Furthermore, DBSCAN was performed on both sets. *Supplementary Table 9* (for the similar set) and *Supplementary Table 10* (for the dissimilar set) summarize the distribution of the training and test data inside and outside the AD determined by the different values of

ADMET Property	Training Set (10-fold CV)		Validation Set (10-fold CV)		Test Set
	ACC	Loss	ACC	Loss	ACC
Ames Test	89.90	0.242	88.93	0.296	89.34
BBB	89.06	0.242	88.63	0.281	91.54
Biodegradation	92.35	0.188	91.16	0.226	92.05
Caco-2	92.67	0.176	91.70	0.176	92.08
Carcinogenicity	98.34	0.049	98.00	0.0.69	97.80
CYP450 1A2 inhibitor	88.18	0.258	87.38	0.303	89.99
CYP450 2C19 inhibitor	91.78	91.25	0.195	0.233	92.35
CYP450 2C9 inhibitor	93.25	92.67	0.163	0.204	95.25
CYP450 2D6 inhibitor	95.21	0.115	94.84	0.141	95.16
CYP450 2D6 substrate	97.17	0.066	97.11	0.090	97.74
CYP450 3A4 inhibitor	91.99	0.188	91.33	0.230	93.51
CYP450 3A4 substrate	92.95	0.173	91.92	0.208	92.94
CYP450 inhibitor promsicuity	92.82	0.178	91.99	0.223	91.34
hERG inhibitor (predictor I)	99.10	0.037	99.13	0.052	99.78
hERG inhibitor (predictor II)	94.07	0.150	93.81	0.184	95.45
Human Intestinal Absorption	94.86	0.130	94.66	0.160	93.43
P-glycoprotein inhibitor I	87.61	0.274	86.82	0.316	87.81
P-glycoprotein inhibitor II	87.41	0.283	86.50	0.343	87.97
P-glycoprotein substrate	92.08	0.197	90.34	0.259	90.98

Table 7. *Statistical results of classification models for ten-fold cross-validated training and validation and test sets for different ADMET endpoints.*

epsilons of DBSCAN. For the similar set, after epsilon = 3, the AD coverage for the test set remains constant, with maximum coverage of 17.89%. For the dissimilar set, however, the AD coverage for the test set increases with the epsilon value, reaching a maximum coverage of 80.56%. This is the opposite of what was expected.

To perform DBSCAN, we first project with PaCMAP. In this case, we used the PaCMAP to project both the training and the external sets. As a result, the projection for the similar set is very similar, and the PaCMap preserves the structure similar to the training set. However, with the dissimilar set, PaCMAP forces the external set to be in the same projection, which could explain why the majority of the compounds are within the AD for this set.

A solution to avoid this is to preserve the projection of the training set and project the external set on it.

3.5 Graph Neural Networks (GNNs)

In this study, 21 GNNs models were implemented, one for each ADMET property. Ten-fold cross-validation and external validation were performed for model validation to ensure that the predictive model had good generalization. In addition, the early stopping function was used to prevent overfitting. This function considers the number of iterations after which the training process will stop if the validation loss does not decrease.

Table 7 summarizes the average training and validation accuracy and loss across the ten-fold cross-validation. Although descriptor- and fingerprint-based models did not perform well for CYP2C9 substrate and hERG inhibitor I, these properties were also included. As expected, the cytochrome enzyme failed to predict; meanwhile, the hERG inhibitor I achieved a test set accuracy of 99.78 and an average validation loss of 0.052. This model, however, is most likely overfitted. The rest of the classification models performed impressively, with an average ACC of 0.93.

In addition, the GNN models for BBB, Caco-2, Carcinogenicity, Ames Test, HIA, and hERG inhibitors (predictor I and II) were validated using an external set. Except for Caco-2 permeable and hERG inhibitor II (ACC=0.49 for both), the models performed admirably. An ACC greater than 0.97 was obtained for the BBB and carcinogenicity. Additionally, 0.9217 and 0.8766 ACC were obtained for the HIA and Ames Test, respectively. Besides giving the model validation, this step also

confirmed that the hERG inhibitor I model was overfitted. The data with approved hERG compounds were predicted using both hERG models (predictor I and II). The ACC obtained for hERG inhibitor II was 0.49, whereas the ACC obtained for hERG inhibitor I was 0. This suggests that balanced data is needed for reliable predictive models.

The model validation was not performed for all the models since obtaining approved molecules for each of the properties is difficult. However, if the validated models gave overall good results, we can assume that the remaining models, with the exception of hERG inhibitor I, give accurate predictions.

3.6 Explicability of the GNN Model

The following attribution methods, IG and Saliency, were used to interpret the predictions of the GNN models for BBB and CYP2C9 inhibitor. The GNNExplainer was then used, and the attribution methods were compared.

As a proof-of-concept and to gain a better understanding of how GNN determines whether a molecule is toxic or not, the important properties' features of ten random CYP2C9 inhibitors and ten non-inhibitors molecules were thoroughly examined. As a result, we see that the highlighted regions for inhibitors correspond to Catechol for almost all of the molecules. One example is shown in **Figure 5**; we can see that two highlighted regions belong to Catechol, a toxic organic compound that acts as a highly reactive radical group.

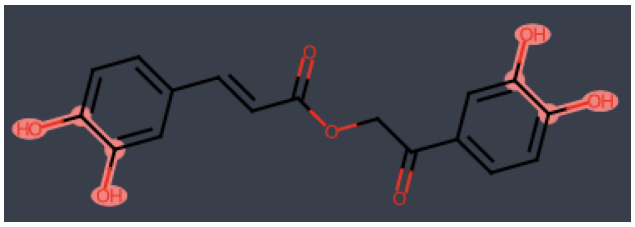

Figure 5. ***Visualizations of attribution scores, calculated using Integrated Gradients.*** Both highlighted regions correspond to Catechol groups.

Furthermore, the important features of BBB were analyzed. The ability of a molecule to cross the blood-brain barrier is determined by factors other than its functional group. Therefore, we cannot draw any conclusions based solely on the highlighted region for BBB. After exploring both importance attribution methods, we can conclude that GNN explicability mapping to the compound topology is more useful for providing further insight on toxicity endpoints.

4 Conclusion

Predicting ADMET properties is critical in drug development because it reduces risks during clinical development. Thus, developing high reliability and computational robustness models to predict these properties is becoming increasingly important. This study reported an extensive ADMET dataset, including 8997 compounds for 21 ADMET points. In order to train the models, the physicochemical properties, as well as various molecular fingerprints, were calculated. Then, several accurate classification and regression models were developed using molecular descriptors and molecular fingerprints with PyCaret. According to the statistical results, the MACCS-based methods outperformed the other ML methods, with remarkable MCC and AUC results for the test set. Furthermore, the applicability domain of the ML models was defined, ensuring the models' predictive ability. After that, GNN-based models were developed and validated, giving good prediction accuracy. Finally, the essential features identified by attribution methods and GNNExplainer for BBB and CYP450 2C9 inhibitor were analyzed to explain the GNN-based models' explicability.

In conclusion, we believe that the models developed in this study can be regarded as simple, accurate, trustworthy, and transparent tools for predicting ADMET properties in drug design and discovery pipelines.

Future Work

Future work will include further refinement of models suspected of having overfitting or unbalancing issues, as well as labeling additional data (from the database assembled in this project) that our models will predict. In addition, we will validate PaCMAP and DBSCAN using a variety of similarity metrics and developing transformation methods around the training set initially projected. Furthermore, additional information about GNNs nodes and edges, such as torsion angles, atom information, chirality, and type, will be included, likely leading to more insight into explainability.

Acknowledgments

I would like to express my deepest appreciation to my supervisor, Alexis Molina, for the patience and excellent guidance throughout the project.

In addition, I would like to thank all of the members of Nostrum Biodiscovery for providing me with an excellent opportunity to work and learn with them.

Supplementary Material

The supplementary material for this project can be found **here**.

File 1. This file contains histograms for the training and test set.

Figure 1. This figure shows the plot for the nearest neighbor function to determine the epsilon value for DBSCAN. **Figure 2.** Scoring plot of the first two principal components in the training and test set for physicochemical property data. **Figure 3.** Scoring plot of the first two principal components in the training and test set for KRFP data. **Figure 4.** Scoring plot of the first two principal components in the training and test set for ECFP2 data. **Figure 5.** Scoring plot of the first two principal components in the training and test set for ECFP4 data. **Figure 6.** Scoring plot of the first two principal components in the training and test set for ECFP6 data. **Figure 7.** Scoring plot of the first two principal components in the training and test set for FCFP2 data. **Figure 8.** Scoring plot of the first two principal components in the training and test set for FCFP4 data. **Figure 9.** Scoring plot of the first two principal components in the training and test set for FCFP6 data.

Table 1. In this table, RDKit-generated atom and bond descriptors used as features for GGN are listed. **Table 2.** The complete list of physicochemical properties obtained. **Table 3.** List of ADMET properties. **Table 4.** List of physicochemical properties used in this study. **Table 5.** The complete performance summary of 189 models. **Table 6.** This table contains the summary of statistical results of the classification and regression model using PCA outliers for MACCS as the test set. **Table 7.** This table contains the summary of statistical results of the classification and regression model using logistic PCA outliers for MACCS as the test set. **Table 8.** This table contains the summary of statistical results of the classification and regression model using DBSCAN outliers for MACCS as the test set. **Table 9.** This table contains the summary of statistical results of the classification and regression model using DBSCAN for the similar set as the test set. **Table 10.** This table contains the summary of statistical results of the classification and regression model using DBSCAN for the dissimilar set as the test set.

References

1. Mullard, A. (2018). 2017 FDA drug approvals. *Nature Reviews Drug Discovery*, *17*(2), 81–85. https://doi.org/10.1038/nrd.2018.4
2. Aleksić, S., Seeliger, D., & Brown, J. B. (2021). ADMET Predictability at Boehringer Ingelheim: State-of-the-Art, and Do Bigger Datasets or Algorithms Make a Difference? *Molecular Informatics*, *41*(2), 2100113. https://doi.org/10.1002/minf.202100113
3. Selick, H. E., Beresford, A. P., & Tarbit, M. H. (2002). The emerging importance of predictive ADME simulation in drug discovery. *Drug Discovery Today*, *7*(2), 109–116. https://doi.org/10.1016/s1359-6446(01)02100-6
4. Kennedy, T. (1997). Managing the drug discovery/development interface. *Drug Discovery Today*, *2*(10), 436–444. https://doi.org/10.1016/s1359-6446(97)01099-4
5. Waring, M. J., Arrowsmith, J., Leach, A. R., Leeson, P. D., Mandrell, S., Owen, R. M., … Weir, A. (2015). An analysis of the attrition of drug candidates from four major pharmaceutical companies. *Nature Reviews Drug Discovery*, *14*(7), 475–486. https://doi.org/10.1038/nrd4609
6. Hay, M., Thomas, D. W., Craighead, J. L., Economides, C., & Rosenthal, J. (2014). Clinical development success rates for investigational drugs. *Nature Biotechnology*, *32*(1), 40–51. https://doi.org/10.1038/nbt.2786
7. Jia, C.-Y., Li, J.-Y., Hao, G.-F., & Yang, G.-F. (2020). A drug-likeness toolbox facilitates ADMET study in drug discovery. *Drug Discovery Today*, *25*(1), 248–258. https://doi.org/10.1016/j.drudis.2019.10.014
8. Kar, S., & Leszczynski, J. (2018). Recent Advances of Computational Modeling for Predicting Drug Metabolism: A Perspective. *Current Drug Metabolism*, *18*(12), 1106–1122. https://doi.org/10.2174/1389200218666170607102104
9. Bhhatarai, B., Walters, W. P., Hop, C. E. C. A., Lanza, G., & Ekins, S. (2019). Opportunities and challenges using artificial intelligence in ADME/Tox. *Nature Materials*, *18*(5), 418–422. https://doi.org/10.1038/s41563-019-0332-5
10. Ferreira, L. L. G., & Andricopulo, A. D. (2019). ADMET modeling approaches in drug discovery. *Drug Discovery Today*, *24*(5), 1157–1165. https://doi.org/10.1016/j.drudis.2019.03.015
11. Kar, S., & Leszczynski, J. (2020). Open access in silico tools to predict the ADMET profiling of drug candidates. *Expert Opinion on Drug Discovery*, *15*(12), 1473–1487. https://doi.org/10.1080/17460441.2020.1798926
12. Wess, G. (2002). How to escape the bottleneck of medicinal chemistry. *Drug Discovery Today*, *7*(10), 533–535. https://doi.org/10.1016/s1359-6446(02)02252-3
13. Xiong, G., Wu, Z., Yi, J., Fu, L., Yang, Z., Hsieh, C., … Cao, D. (2021). ADMETlab 2.0: an integrated online platform for accurate and comprehensive predictions of ADMET properties. *Nucleic Acids Research*, *49*(W1), W5–W14. https://doi.org/10.1093/nar/gkab255
14. Daina, A., Michielin, O., & Zoete, V. (2017). SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific Reports*, *7*(1). https://doi.org/10.1038/srep42717
15. Yang, H., Lou, C., Sun, L., Li, J., Cai, Y., Wang, Z., … Tang, Y. (2018). admetSAR 2.0: web-service for prediction and optimization of chemical ADMET properties. *Bioinformatics*, *35*(6), 1067–1069. https://doi.org/10.1093/bioinformatics/bty707
16. Banerjee, P., Eckert, A. O., Schrey, A. K., & Preissner, R. (2018). ProTox-II: a webserver for the prediction of toxicity of chemicals. *Nucleic Acids Research*, *46*(W1), W257–W263. https://doi.org/10.1093/nar/gky318
17. Schyman, P., Liu, R., Desai, V., & Wallqvist, A. (2017). vNN Web Server for ADMET Predictions. *Frontiers in Pharmacology*, *8*. https://doi.org/10.3389/fphar.2017.00889
18. Pires, D. E. V., Blundell, T. L., & Ascher, D. B. (2015). pkCSM: Predicting Small-Molecule Pharmacokinetic and Toxicity Properties Using Graph-Based Signatures. *Journal of Medicinal Chemistry*, *58*(9), 4066–4072. https://doi.org/10.1021/acs.jmedchem.5b00104
19. Patel, R. D., Prasanth Kumar, S., Pandya, H. A., & Solanki, H. A. (2018). MDCKpred: a web-tool to calculate MDCK permeability coefficient of small molecule using membrane-interaction chemical features. *Toxicology Mechanisms and Methods*, *28*(9), 685–698. https://doi.org/10.1080/15376516.2018.1499840
20. Zhang, L., Ai, H., Chen, W., Yin, Z., Hu, H., Zhu, J., … Liu, H. (2017). CarcinoPred-EL: Novel models for predicting the carcinogenicity of chemicals using molecular fingerprints and ensemble learning methods. *Scientific Reports*, *7*(1). https://doi.org/10.1038/s41598-017-02365-0
21. Wang, Y.-W., Huang, L., Jiang, S.-W., Li, K., Zou, J.,, & Yang, S.-Y. (2020). CapsCarcino: A novel sparse data deep learning tool for predicting carcinogens. *Food and Chemical Toxicology*, *135*, 110921. https://doi.org/10.1016/j.fct.2019.110921
22. Yap, C. W. (2010). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry*, *32*(7), 1466–1474. https://doi.org/10.1002/jcc.21707

23. Venkatraman, V., & Alsberg, B. K. (2016). KRAKENX: software for the generation of alignment-independent 3D descriptors. *Journal of Molecular Modeling*, *22*(4). https://doi.org/10.1007/s00894-016-2957-5
24. Hanser, T., Barber, C., Marchaland, J. F., & Werner, S. (2016). Applicability domain: towards a more formal definition. *SAR and QSAR in Environmental Research*, *27*(11), 865–881. https://doi.org/10.1080/1062936x.2016.1250229
25. Netzeva, T. I., Worth, A. P., Aldenberg, T., Benigni, R., Cronin, M. T. D., Gramatica, P., … Veith, G. (2005). Current Status of Methods for Defining the Applicability Domain of (Quantitative) Structure-Activity Relationships. *Alternatives to Laboratory Animals*, *33*(2), 155–173. https://doi.org/10.1177/026119290503300209
26. Gadaleta, D., Mangiatordi, G. F., Catto, M., Carotti, A., & Nicolotti, O. (2016). Applicability Domain for QSAR Models. *International Journal of Quantitative Structure-Property Relationships*, *1*(1), 45–63. https://doi.org/10.4018/ijqspr.2016010102
27. Sahigara, F., Mansouri, K., Ballabio, D., Mauri, A., Consonni, V., & Todeschini, R. (2012). Comparison of Different Approaches to Define the Applicability Domain of QSAR Models. *Molecules*, *17*(5), 4791–4810. https://doi.org/10.3390/molecules17054791
28. Gawehn, E., Hiss, J. A., & Schneider, G. (2015). Deep Learning in Drug Discovery. *Molecular Informatics*, *35*(1), 3–14. https://doi.org/10.1002/minf.201501008
29. Goh, G. B., Hodas, N. O., & Vishnu, A. (2017). Deep learning for computational chemistry. *Journal of Computational Chemistry*, *38*(16), 1291–1307. https://doi.org/10.1002/jcc.24764
30. Elton, D. C., Boukouvalas, Z., Fuge, M. D., & Chung, P. W. (2019). Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, *4*(4), 828–849. https://doi.org/10.1039/c9me00039a
31. Torng, W., & Altman, R. B. (2019). Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *Journal of Chemical Information and Modeling*, *59*(10), 4131–4149. https://doi.org/10.1021/acs.jcim.9b00628
32. Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., & Svetnik, V. (2015). Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling*, *55*(2), 263–274. https://doi.org/10.1021/ci500747n
33. Schneckener, S., Grimbs, S., Hey, J., Menz, S., Osmers, M., Schaper, S., … Göller, A. H. (2019). Prediction of Oral Bioavailability in Rats: Transferring Insights from in Vitro Correlations to (Deep) Machine Learning Models Using in Silico Model Outputs and Chemical Structure Parameters. *Journal of Chemical Information and Modeling*, *59*(11), 4893–4905. https://doi.org/10.1021/acs.jcim.9b00460
34. Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., … Barzilay, R. (2019). Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, *59*(8), 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237
35. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. *ArXiv.org*. https://doi.org/10.48550/arXiv.1704.01212
36. Kearnes, S., McCloskey, K., Berndl, M., Pande, V., & Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, *30*(8), 595–608. https://doi.org/10.1007/s10822-016-9938-8
37. Shang, C., Liu, Q., Chen, K.-S., Sun, J., Lu, J., Yi, J., & Bi, J. (2018). Edge Attention-based Multi-Relational Graph Convolutional Networks. *ArXiv.org*. https://doi.org/10.48550/arXiv.1802.04944
38. Li, J., Cai, D., & He, X. (2017). Learning Graph-Level Representation for Drug Discovery. *ArXiv.org*. https://doi.org/10.48550/arXiv.1709.03741
39. Wu, Z., Ramsundar, B., Feinberg, Evan N., Gomes, J., Geniesse, C., Pappu, A. S., … Pande, V. (2018). MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, *9*(2), 513–530. https://doi.org/10.1039/c7sc02664a
40. Korolev, V., Mitrofanov, A., Korotcov, A., & Tkachenko, V. (2019). Graph Convolutional Neural Networks as "General-Purpose" Property Predictors: The Universality and Limits of Applicability. *Journal of Chemical Information and Modeling*, *60*(1), 22–28. https://doi.org/10.1021/acs.jcim.9b00587
41. Withnall, M., Lindelöf, E., Engkvist, O., & Chen, H. (2020). Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. *Journal of Cheminformatics*, *12*(1). https://doi.org/10.1186/s13321-019-0407-y
42. Hop, P., Allgood, B., & Yu, J. (2018). Geometric Deep Learning Autonomously Learns Chemical Features That Outperform Those Engineered by Domain Experts. *Molecular Pharmaceutics*, *15*(10), 4371–4377. https://doi.org/10.1021/acs.molpharmaceut.7b01144
43. Jin, W., Coley, C. W., Barzilay, R., & Jaakkola, T. (2017). Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. *ArXiv.org*. https://doi.org/10.48550/arXiv.1709.04555
44. Jiménez-Luna, J., Grisoni, F., & Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, *2*(10), 573–584. https://doi.org/10.1038/s42256-020-00236-4
45. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., … Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9
46. Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. *ArXiv.org*. https://doi.org/10.48550/arXiv.1903.03894
47. Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., & Overington, J. P. (2011). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, *40*(D1), D1100–D1107. https://doi.org/10.1093/nar/gkr777
48. Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, *35*(Database), D198–D201. https://doi.org/10.1093/nar/gkl999
49. Williams, A. J., Grulke, C. M., Edwards, J., McEachran, A. D., Mansouri, K., Baker, N. C., … Richard, A. M. (2017). The CompTox Chemistry Dashboard: a community data resource for environmental chemistry. *Journal of Cheminformatics*, *9*(1). https://doi.org/10.1186/s13321-017-0247-6
50. Mohanraj, K., Karthikeyan, B. S., Vivek-Ananth, R. P., Chand, R. P. B., Aparna, S. R., Mangalapandi, P., & Samal, A. (2018). IMPPAT: A curated database of Indian Medicinal Plants, Phytochemistry And Therapeutics. *Scientific Reports*, *8*(1). https://doi.org/10.1038/s41598-018-22631-z
51. Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., … Bolton, E. E. (2020). PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, *49*(D1), D1388–D1395. https://doi.org/10.1093/nar/gkaa971
52. Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., … Hassanali, M. (2007). DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, *36*(suppl_1), D901–D906. https://doi.org/10.1093/nar/gkm958
53. Schneider, G. (2010). Virtual screening: an endless staircase? *Nature Reviews Drug Discovery*, *9*(4), 273–276. https://doi.org/10.1038/nrd3139
54. RDKit: Open-source cheminformatics; http://www.rdkit.org
55. Yang, Z.-Y., Yang, Z.-J., Lu, A.-P., Hou, T.-J., & Cao, D.-S. (2020). Scopy: an integrated negative design python library for desirable HTS/VS database design. *Briefings in Bioinformatics*, *22*(3). https://doi.org/10.1093/bib/bbaa194
56. Lo, Y.-C., Rensi, S. E., Torng, W., & Altman, R. B. (2018). Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today*, *23*(8), 1538–1546. https://doi.org/10.1016/j.drudis.2018.05.010
57. *The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service.* (2022). ACS Publications. https://pubs.acs.org/doi/abs/10.1021/c160017a018

58. Riniker, S., & Landrum, G. A. (2013). Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of Cheminformatics*, *5*(1). https://doi.org/10.1186/1758-2946-5-26
59. Ji, H., Deng, H., Lu, H., & Zhang, Z. (2020). Predicting a Molecular Fingerprint from an Electron Ionization Mass Spectrum with Deep Neural Networks. *Analytical Chemistry*, *92*(13), 8649–8653. https://doi.org/10.1021/acs.analchem.0c01450
60. Lavecchia, A. (2015). Machine-learning approaches in drug discovery: methods and applications. *Drug Discovery Today*, *20*(3), 318–331. https://doi.org/10.1016/j.drudis.2014.10.012
61. Schroeter, T. S., Schwaighofer, A., Mika, S., Ter Laak, A., Suelzle, D., Ganzer, U., … Müller, K.-R. (2007). Estimating the domain of applicability for machine learning QSAR models: a study on aqueous solubility of drug discovery molecules. *Journal of Computer-Aided Molecular Design*, *21*(9), 485–498. https://doi.org/10.1007/s10822-007-9125-z
62. Wold, S., Sjöström, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, *58*(2), 109–130. https://doi.org/10.1016/s0169-7439(01)00155-1
63. Ester, M., Kriegel, Hans-P., Sander, J, & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). *AAAI Press*, 226–231.https://dl.acm.org/doi/10.5555/3001460.3001507
64. Wang, Y., Huang, H., Rudin, C., & Shaposhnik, Y. (2012). Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization. *ArXiv.org*. https://doi.org/10.48550/arXiv.2012.04456
65. Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, *6*(1). https://doi.org/10.1186/s40649-019-0069-y
66. Fey, M., & Lenssen, J. E. (2019). Fast Graph Representation Learning with PyTorch Geometric. *ArXiv.org*. https://doi.org/10.48550/arXiv.1903.02428
67. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Bai, J. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *ArXiv.org*. https://doi.org/10.48550/arXiv.1912.01703
68. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Retrieved June 15, 2022, from arXiv.org website: https://arxiv.org/abs/1412.6980v9
69. Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. *ArXiv.org*. https://doi.org/10.48550/arXiv.1703.01365
70. Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. *ArXiv.org*. https://doi.org/10.48550/arXiv.1903.03894