Artificial Intelligence Chemistry 1 (2023) 100020

Contents lists available at ScienceDirect

Artificial Intelligence Chemistry

journal homepage: www.journals.elsevier.com/artificial-intelligence-chemistry

Machine learning small molecule properties in drug discovery

Nikolai Schapin [a,b,*], Maciej Majewski [a], Alejandro Varela-Rial [a], Carlos Arroniz [a], Gianni De Fabritiis [b,c,d,*]

[a] Acellera Labs, C/ Doctor Trueta 183, 08005 Barcelona, Spain
[b] Computational Science Laboratory, Universitat Pompeu Fabra, PRBB, C/ Doctor Aiguader 88, 08003 Barcelona, Spain
[c] Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluís Companys 23, 08010 Barcelona, Spain
[d] Acellera, Devonshire House 582 Honeypot Lane Stanmore, Middlesex HA7 1JS United Kingdom

ARTICLE INFO

Keywords:
Molecular property prediction
ADMET prediction models
Binding affinity prediction models
Physicochemical properties prediction models
Computational methods in drug discovery

ABSTRACT

Machine learning (ML) is a promising approach for predicting small molecule properties in drug discovery. Here, we provide a comprehensive overview of various ML methods introduced for this purpose in recent years. We review a wide range of properties, including binding affinities, solubility, and ADMET (Absorption, Distribution, Metabolism, Excretion, and Toxicity). We discuss existing popular datasets and molecular descriptors and embeddings, such as chemical fingerprints and graph-based neural networks. We highlight also challenges of predicting and optimizing multiple properties during hit-to-lead and lead optimization stages of drug discovery and explore briefly possible multi-objective optimization techniques that can be used to balance diverse properties while optimizing lead candidates. Finally, techniques to provide an understanding of model predictions, especially for critical decision-making in drug discovery are assessed. Overall, this review provides insights into the landscape of ML models for small molecule property predictions in drug discovery. So far, there are multiple diverse approaches, but their performances are often comparable. Neural networks, while more flexible, do not always outperform simpler models. This shows that the availability of high-quality training data remains crucial for training accurate models and there is a need for standardized benchmarks, additional performance metrics, and best practices to enable richer comparisons between the different techniques and models that can shed a better light on the differences between the many techniques.

1. Introduction

Early stage, preclinical drug discovery is a step-wise process, where at each stage, hit molecules are required to meet certain criteria to ensure their efficacy and quality before proceeding to the next stage. This results in a series of molecular properties that need to be optimized. In order to do this, they need to be measured, which traditionally is done through wet-lab experiments that are costly and time-consuming. The estimated R&D expenditure is around 41 billion euros in Europe and 83 billion US dollars in the USA with R&D costs per drug ranging around 1–2 billion US dollars [1,2]. This process takes on average 10–13 years [1,2], and only 1 out of 10 000 substances tested [2] will pass through all the stages to become a new successfully marketed drug. While most of the cost and two-thirds of the time are linked to the stages of clinical trials, most of the candidate molecules fail during these stages [3]. The main reasons [4] for failure are low efficacy of the drug, high toxicity, or commercial reasons. The first two are often the result of unsuccessful or insufficient establishment of key molecular properties of the hit and lead candidates during the early stages of drug discovery.

Various resource-efficient computational techniques have been developed which all fall under the group of computer-aided drug design methods [5] in order to improve the initial screening of compounds in these early stages by both increasing the amount of screened compounds and enabling compound selection and prioritization. These various methods use computational algorithms and models to estimate the molecular properties of screened compounds without having to recur to expensive laboratory experimentation.

Various groups of methods have been developed such as docking algorithms, molecular dynamics algorithms, quantum mechanical/molecular mechanical (QM/MM) simulations and empirical scoring functions. Docking algorithms [6] provide a way to compute binding poses of ligands to their targets by using interaction information between both and generating a score. This score can also be used to approximate binding affinities, however, it has been shown to produce low accuracy

* Corresponding authors at: Computational Science Laboratory, Universitat Pompeu Fabra, PRBB, C/ Doctor Aiguader 88, 08003 Barcelona, Spain.
E-mail addresses: n.shapin@acellera.com (N. Schapin), g.defabritiis@acellera.com (G.D. Fabritiis).

https://doi.org/10.1016/j.aichem.2023.100020
Received 30 July 2023; Received in revised form 19 September 2023; Accepted 17 October 2023
Available online 19 October 2023

results [7,8].

More accurate estimates of binding affinity can be obtained with molecular dynamics or QM/MM simulations [9] that simulate atom and molecular movements using computations to approximate the force field that drives this atomic movement. Binding affinities can then be estimated from the forces acting on the molecular structures. A drawback of these methods is that they are computationally expensive, making it difficult to analyze large amounts of compounds. A faster approach is the use of empirical scoring functions [10]. This approach involves the application of molecular or structural descriptors of screened compounds, such as quantitative structure-activity relationship (QSAR) or quantitative structure-property relationship (QSPR) models, or representations of protein-ligand complexes together with either fixed functional form models or more complex machine learning models to predict different molecular properties. Fixed functional form models can lack accuracy as they use manually derived rules which can fail to model complex relationships found in the input data. Machine learning models can improve on this by using more flexible models that can model both linear and non-linear relationships. Furthermore, as they learn strictly from the observations, human construction bias can be avoided. Various machine learning methods [11] have therefore been developed for different tasks within the drug discovery process.

In this review paper, we will take a look at the current role of machine learning in predicting specifically molecular properties of small molecules for drug discovery. We will analyze how machine learning techniques have been applied so far to determine various important molecular properties. We will assess whether they were able to reach enough accuracy and speed and whether they found a practical implementation in drug discovery. We will also comment on the potential disadvantages of these techniques and their causes. Lastly, we will formulate conclusions on the current state of these techniques in drug discovery and suggest some interesting areas for future research in this field.

2. Model types

Different types of machine learning models with varying degrees of complexity can predict molecular properties such as similarity-based models, linear models, kernel-based models, Bayesian models, tree-based models, and neural networks.

These types of models can generally be classified into non-neural network models and more complex neural network models. Among the non-neural network models one can further distinguish models that do not fit any specific functional form such as similarity-based models and those that fit one based on the training data. Similarity-based models try to generate predictions for provided compounds based on known molecular properties of similar compounds. Among the models that do fit a functional form to training data one can further distinguish various functional forms. A first functional form is linear functions that are employed in linear and kernel-based models. Linear models use linear functional forms to learn linear relationships between representations of input molecular structures and their molecular properties. Kernel-based models transform the input data using kernel functions in order to improve the predictions of the data. More flexible approaches include Bayesian models where the exact functional form can be loosely defined such that one would fit a function over multiple functional forms. Bayesian models use the logic of Bayesian inference where they fit a function of loose form to the data in order to minimize the difference between the obtained likelihood distribution of model predictions and the true posterior distribution of the data. This loose functional form can be also defined more exactly by using a specific functional form whereby the fitting would then be performed over the function's hyperparameters only. Another functional form that can be used is decision trees which are characterized by nodes that represent learned decision rules, branches which are the outcomes of these decision rules, and that can further branch off into deeper sub-branches and leaf nodes that represent outcomes when following a specific path along the tree. During fitting, one tries to learn the decision rules necessary to separate the training data into classes or value ranges. The tree-based models can make use of single or multiple decision trees constructed from a selection of features from the input structures. Lastly, neural networks, being the other large group of ML models, employ models that consist of many interconnected neurons organized in layers that perform non-linear transformations and aggregations of the input data to learn complex, non-linear relationships. Depending on the organization of the neurons and the type of format of the input data, neural network models can further be divided among feed-forward neural network models and graph-based models. A more detailed description and examples for each type are discussed and presented in the following subsections.

2.1. Similarity-based models

The first group describes a simple and straightforward technique which is the k-nearest neighbors (kNN) technique [12]. This method is based on similarities between the data points and uses the principle that similar data points will also yield a similar outcome value. It performs predictions about new data points by applying a weighted average between the k-nearest neighbors of that data point where k can be a user-defined integer value. One can use different similarity measures to compute the similarities. Closely related, the nearest centroid method [13] uses the closest distance to the center of the established clusters instead. As the latter method uses distances to cluster center points rather than individual most similar data points, it is more suited for classification tasks rather than regression.

The simplicity of these techniques and the absence of a learnable functional form make that they can be easily used without any prior training on data. By using dimensionality reduction techniques such as t-SNE [14] for multi-dimensional data one can also obtain visualizations of the defined clusters and inter-datapoint distances which can help to easily explain predictions on new datapoints. On the other hand, the simplicity of not having a functional form makes that the technique relies solely on similarities between datapoints to make its predictions. This means that complex correlations between the datapoints cannot be learned by the model which can result in underfitting with poor performance on unseen data.

2.2. Linear models

Different types of linear models exist which mainly differ in the ways they fit the linear functions to the input data. Linear models and multivariate linear models in the case of multiple variables, work by fitting a linear function to the data in order to either establish a regression function like multivariate linear regression (MLR) or a linear classification boundary. The data itself is presented as a feature vector and the resulting fitted linear function will have the same dimension as the input data feature vector. Flexible discriminant analysis [15] can be seen as an extension of linear models for multi-class classification problems. It uses non-parametric regression as opposed to classic linear discriminant analysis to find groups of data and applies further classic linear discriminant analysis to maximize the separation of the data between the found groups. Multivariate adaptive regression splines (MARS) [16] is another extension of linear models where the full dataset is split into chunks and where multiple individual linear models are constructed for each chunk.

When constructing computational models, one needs to be aware of model overfitting to the training data. Models that overfit become too tailored to the training data which reduces their generality and performance on new unseen data. Several extensions to linear models exist that can solve this for linear models. Lasso regression [17] is one such technique. It imposes a regularization term on the number of coefficients to avoid overfitting. In particular, it uses L1 regularization by adding the absolute value of the magnitude of the coefficient as a penalty term which, when learned, shrinks the coefficients of the less important features to zero. On the contrary, ridge regression [18] applies an L2 regularization penalty by adding the square of the magnitude of the weights.

Partial Least Squares (PLS) [19] regression is another linear method for regression problems which is especially useful when the number of predictors is higher than the number of observations, a common situation in cheminformatics, and when there is multicollinearity between the input variables. PLS tries to reduce the number of input variables to a subset that maximally is able to explain the correlation with the observed values.

While the absence of a learnable functional form in similarity-based models (see Section 2.1) makes it impossible for them to learn more complex correlations in the data, the introduction of such a functional form in linear models makes it possible for them to use learned linear relationships in downstream predictions. As with all learnable functions, one needs to be aware of overfitting which can be resolved for these models by techniques such as Lasso and ridge. The simplicity of the linear functional form however can result in underfitting of these models in data where the desired predicted properties depend on more complex, non-linear relationships which often are frequent among biomedical and molecular data.

2.3. Kernel-based models

One obvious mentioned drawback of linear models is their inability to fit non-linear functions to the data, which can reduce their performance on data that consists of complex, non-linear relationships between its input features and the to-be-predicted molecular properties. Several models exist that employ kernel tricks that transform the input data in such a way that it can improve the fit of linear models on these transformed data points. Support vector machines (SVMs) [20] do this by transforming the input data into higher dimensions where they become linearly separable. This transformation is achieved through kernel functions like radial basis functions. Originally SVMs were defined for classification problems [20] but were later expanded to regression problems as well [21]. Kernel ridge regression [22], applies a similar kernel trick as in SVM to the input data but uses ridge regression to construct the linear model. While both seem to be very similar, the difference lies in the construction of the linear model. In the case of ridge regression, the construction is done by fitting the data to the linear function while in SVM it is based on the use of support vectors, which are minimal points in the data that allow describing the optimal separator function of the data [23]. Another simple way to fit data to output values is by using radial basis functions [24] to approximate the unknown function.

The use of the kernel trick makes it possible to use simple linear functions which are easier to interpret and have less possibility of overfitting due to the restricted number of learnable parameters and the linear functional form on data where the molecular properties of interest are governed by non-linear relationships with their molecular descriptors. The type of kernel function used is hereby crucial as it will determine how easily the transformed input data can be modelled through linear functions. Furthermore, the type and complexity of the kernel function will also influence how easily the model can be interpreted as this is harder for more complex, non-linear kernel functions.

2.4. Bayesian models

Another type of method that allows modeling non-linear relationships and does not involve more complex neural network models are Gaussian processes [25]. These models apply principles from Bayesian inference and assume a prior probability distribution of the values of the unknown function that the Gaussian process wants to model. This distribution is updated to fit the obtained likelihood to the posterior distribution. Different from linear and kernel-based models, where a set of random parameters are fit to a fixed function form, Gaussian processes allow modeling the prior probability distribution of the model function over all possible functional forms and their parameters. A case of Bayesian models where the functional form is fixed is the maximum likelihood estimation which uses Bayesian inference to estimate only the parameters of a chosen fixed-form probability distribution.

One important major advantage of these models is that by design they are able to provide uncertainty estimates on the generated predictions. While it is possible to obtain such estimates with other ML models through various techniques [26] such as frequentist methods like similarity-based methods or Platt Scaling or some Bayesian approaches like training model ensembles, Monte Carlo dropout or bootstrapping, pure Bayesian ML models have such error estimations natively included. Such error estimations are very important for ML models used in drug discovery for several reasons. New ligands can often be out-of-distribution from the data used to train the models where model generalizability on out-of-distribution data might be poor. ML models are also employed to guide decision-making in the different stages of drug discovery and can also be incorporated into active learning cycles where the ML models can be further fine-tuned with experimentally predicted datapoints for which high errors were predicted for their predicted values of molecular properties. One needs however to pay attention to prior selection when using Bayesian models as this can affect the model's performance. Especially when using informative priors it is important that these remain applicable to the data.

2.5. Tree-based models

Decision tree models [27–30,31] fit a decision tree to the training data which also consists of a feature vector of fixed size. The model uses hereby the features to construct tree nodes and tries to fit decision rules to build up the branches of the decision tree. While a single decision tree can fit all the training data, the performance might be improved when using an ensemble of decision trees each fitted to only a subset of the training data. This is what random forest models [32] do. The final prediction they generate is constructed from a weighted average of the single predictions of each decision tree in the forest. XGBoost, boosted trees and gradient boosted tree models [33,34] further improve the technique of random forest models by fitting each tree sequentially instead of in parallel like in random forests and using information from the existing trees to improve the performance of the following ones. Because the performance of random forests depends on the strength of each individual decision tree [32], this way of constructing the trees often gives improved performance over classical random forests.

Decision tree-based methods' predictions are usually easier to interpret as decision boundaries are constructed directly from input features of the data. One can therefore highlight which features received more weight from the model when generating the predictions (see Section 4.5). Various tree-based methods, in particular random forest and XGBoost models, showed also good results for various molecular properties prediction tasks such as binding affinity prediction or estimation of physicochemical and ADMET properties, performing even on-par or better than more complex neural network models (see Section 4.3.2). A crucial aspect, which likewise also applies to the other ML methods is the featurization of the input data. As this directly influences the performance of the ML models and can allow complex 3D input data (see Section 4.3.2) to be fitted by more simple models, such as XGBoost, without the need to resort to more complex graph-based neural network models. This can be a good solution when little training data is available to fit a model without overfitting the training data.

2.6. Neural networks

Finally, the last group of ML models used for molecular properties prediction are neural network models. These models tend to be larger and more complex than the previously presented models and are also highly non-linear. The basic building blocks of neural network models are neurons that can be organized and connected in different ways, which gives a large variety of different types of neural network models such as feed-forward neural networks and graph neural networks.

2.6.1. Feed-forward neural networks

One of the simplest neural network models that can be used for molecular property predictions is feed-forward neural networks, such as a multilayer perceptron (MLP) [35]. They consist of neurons which are the building blocks of neural networks and which transform the information according to the general formula

$$y = \sigma(x * W + b)$$

with x and y being the input and output of each neuron respectively, W and b the weight and bias assigned to each connection between neurons and σ an activation function applied to the neuron's output which can be either a linear or non-linear function. These neurons are further organized in layers. The input layer consists of an equal number of neurons as the number of features in the input data vector and is used to receive the input data. This is followed by one or multiple hidden layers which can be larger or smaller in size but which usually are densely connected, meaning that each neuron in a layer is connected to both all neurons of the previous and next layers. The final layers usually consist of a single output neuron that gives the scalar prediction in case of regression problems or a probability in case of binary classification. It can also consist of an output layer with multiple neurons in case of multi-class classification which gives per-class probabilities. Deep neural networks are feed-forward neural networks with a large number of hidden layers. Just as in linear models, they can contain additional tricks to improve their performance and reduce overfitting such as skip-connections, where unperturbed inputs are propagated together with their transformed counterparts to mitigate the risk of vanishing gradients, batch normalization to avoid large differences in the weights of each layer or dropout layers to avoid overfitting where a random selection of neurons are not used in specific layers.

2.6.2. Graph neural networks

So far, all of the described ML models were using 2D feature vectors to represent the molecular structural data. Graph neural networks (GNN) are a group of neural network models that can use either 2D or 3D molecular representations depending on how the input structures are presented, allowing them to explicitly use structural bond information between the atoms and 3D conformational information in the case of 3D representations. In this group of graph-based models, we can distinguish several types such as graph convolutional networks (GCN) [36], graph attention networks (GAT) [37], or message passing neural networks (MPNN) [38]. All of these architectures model the molecular data as a 2D or 3D graph made of nodes and edges that represent atoms and bonds or interatomic distances respectively. Node and edge information is hereby represented as feature vectors of fixed size. The main difference between the different graph-based models is how they combine the information of nodes and edges in the molecular graph.

Graph convolutional neural networks: Graph convolutional neural networks do this through the following propagation rule:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^{(l)}W^{(l)})$$

with $\widetilde{A} = A + I_N$ being the adjacency matrix of the graph with added self connections. The adjacency matrix is a square 2D matrix with rows and columns equal to the number of nodes in the graph and it describes the connectivity between the nodes. $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$, $W^{(l)}$ is the weight matrix for each layer and $H^{(l)}$ is the matrix of neuron activations. $\sigma(\cdot)$ is the activation function. Hereby node embeddings are updated with information from other nodes in the graph taking their graph connectivity into account. This allows the network to learn non-local, non-linear relationships by stacking multiple graph convolutional layers.

Graph attention networks: In graph attention networks, the node update operation involves an attention mechanism different from the GCNs. Concretely, the propagation rule is:

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}W\vec{h}_j\right)$$

where $\vec{h}'_i$ are the updated node embeddings, σ is the applied non-linearity, α_{ij} are the learnable attention weights, W are the weights of the neurons, $\vec{h}_j$ the node embeddings of the current and other nodes in the graph and $\Vert$ a concatenation across multiple attention blocks which showed better performance than when using single attention. These attention mechanisms allow the model to better learn important node connections and relations within the molecular graph and can further be manually customized by, for example, incorporating masks to only focus on local neighborhood information.

Message-passing neural networks: Lastly, message-passing neural networks instead update their node information based on neighboring nodes through message functions. Generally, message functions can be formulated as

$$m_v^{t+1} = \sum_{w\in\mathcal{N}(v)} M_t\left(h_v^t, h_w^t, e_{vw}\right)$$

with m_v^{t+1} being the total message obtained as a sum of messages coming from all neighboring nodes, M_t the individual message function operating between node embeddings of the central node h_v^t and each of its neighbors h_w^t, where their connection is specified by e_{vw}. These message functions are layers with learnable parameters and non-linearity and can be further customized with masks or different ways of single message aggregation. The node embeddings are then updated according to

$$h_v^{t+1} = U_t\left(h_v^t, m_v^{t+1}\right)$$

where U_t is an update function with learnable weights and non-linearity that combines the node embeddings of each node with the message generated from their local neighborhood. These models use only local nodes that are immediately connected or that are within a predefined cutoff from the central node.

Advantages and Considerations: Neural network models are generally highly non-linear models when using non-linear activation functions and through their architectural flexibility can approximate any functional form. This flexibility and the possibility to create models with a large number of parameters can also lead to a higher risk for overfitting on the training data, creating the need for a larger amount of datapoints during training or L1 or L2 regularization techniques to reduce overfitting. On the other hand, the architectural flexibility does also allow to construct models for a diverse range of input data, ranging from 2D vector inputs, 2D images and graphs to more complex 3D volumetric or graph representations. It also allows the construction of more generalizable models for data with multiple output values, such as multi-protic compounds in pKa prediction models, or makes use of combinations of multiple correlated datasets for different molecular properties as a way to increase the amount of training data available.

3. Datasets

In ML model training, the foundation of success lies in the utilization of extensive and top-tier training data for each specific molecular property. These invaluable datasets can be sourced either from internal, proprietary repositories or harnessed from the vast expanse of publicly available resources. In this section, we offer a comprehensive overview of curated public datasets that cater to various essential molecular properties Table 1.

Table 1
Overview of popular datasets for molecular properties information used to train many ML applications.

Dataset name	Type of molecule	Source of molecular structure	Source of measurement
Mixed Datasets			
DrugBank	Small molecules	SMILES	Experimental and predicted
CHEMBL	Small molecules	SMILES, target identifier, PDB ID of crystal	Experimental and predicted
PubChem	Small molecules	SMILES	Experimental and predicted
Therapeutic Data Commons	Small molecules, complexes and targets	SMILES, 3D structures, crystals, docking or MD	Experimental & assumed
Binding Affinity Datasets			
PDBBind	Complexes	Crystals	Experimental
BindingDB	Complexes	Crystals & docking	Experimental
Binding MOAD	Complexes	Crystals	Experimental
PLAS-5k [39]	Complexes	Crystals & MD	Experimental
MISATO [40]	Complexes	Crystals & MD	Experimental
KIBA	Kinases	Target sequences & ligand SMILES	Experimental
Dud-e	Targets, actives & decoys	3D structures	Experimental & assumed
MUV	Targets, actives & decoys	3D structures	Experimental & assumed
LIT-PCBA	Complexes	Docking	Experimental
Physicochemical and ADMET Datasets			
PHYSPROP	Small molecules	SMILES	Experimental
logP dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
Lipophilicity dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
logS dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
FreeSolv 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
ESOL	Small molecules	SMILES	Experimental
BBBP dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
Quantitative toxicity dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
DAT dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
hERG dataset 2D/3D	Small molecules	SMILES & 3D coordinates	Experimental
AMES dataset	Small molecules	SMILES	Experimental
Tox21	Small molecules	SMILES	Experimental
ToxCast	Small molecules	SMILES	Experimental
ClinTox [41]	Small molecules	SMILES	Experimental

Abbreviations: MD = molecular dynamics

3.1. Mixed datasets

Several, large-scale datasets, such as DrugBank, CHEMBL, PubChem, and Therapeutics Data Commons (TDC) exist that host data on multiple molecular properties important for drug discovery.

DrugBank [42] is a dataset focusing specifically on commercially available registered drugs and their targets. The current version contains 15790 drugs of which the majority are small molecules and 3392 biologics. They further provide also information on the drug's commercial availability status, its mode of action, and physicochemical and ADMET properties. CHEMBL [43] is a major dataset of bioactive molecular data with drug-like properties with information on various molecular properties and assays such as binding assays, ADME endpoints, toxicology information, and physicochemical properties like pKa, solubility, and lipophilicity. The dataset contains in total around 2.4 million compounds and various amounts of assay data: 478978 binding information datapoints, 280586 ADME datapoints, 50784 toxicity datapoints and 24290 physicochemical assay datapoints. Differently from the binding datasets, CHEMBL contains the SMILES of the ligands and an identifier of the protein target or a reference to the protein-ligand complex on PDB. PubChem [44] is another major dataset of around 115 million compounds with 305 million recorded bioactivity and toxicity information. Therapeutics Data Commons (TDC) [45,46] is an initiative and platform developed to facilitate the creation of new ML tools in various therapeutic areas. To enable this, the TDC holds in total 15919332 datapoints across 66 various datasets curated and prepared for ML model construction across 22 different prediction tasks. Apart from that, they provide additional data split functions, molecular generation algorithms, and data processing tools and hold various leaderboards to compare publicly available models and techniques on standardized benchmarks.

3.2. Binding affinity datasets

The various datasets hosting binding affinity data can generally be divided into two groups, datasets that hold continuous binding affinity values for each protein-ligand complex and datasets based on a binary classification between binding and non-binding ligands to their targets. Datasets of the first group are PDBBind, BindingDB, Binding MOAD, KIBA, PLAS-5k and MISATO. In the second group, we have the Dud-e, Maximum Unbiased Validation (MUV) and LIT-PCBA datasets.

3.2.1. Datasets for binding affinities

PDBBind [47], BindingDB [48] and binding MOAD [49] were the first datasets containing protein-ligand complexes and experimental binding affinity values. All three have a certain degree of overlap and operate on curated subsets of the Protein Data Bank (PDB) database.

The latest version of PDBBind currently holds 23496 complexes out of which 19443 protein-ligand complexes. This set is again divided into two parts, the general and refined sets. The refined set is a higher quality subset that was curated using a range of filters: (1) Inclusion of compounds with a resolution < = 2.5 Å and an R-factor < = 0.250, (2) Exclusion of ligands with covalent bonds to the target, (3) Exclusion of complexes with multiple ligands bound in the same active site, (4) Exclusion of complexes with steric clashes (< 2.0 Å) between ligand and protein, (5) Exclusion of complexes where the ratio of the buried solvent-accessible surface of the ligand exceeds 15%, (6) Exclusion of complexes with non-standard residues that are in direct contact with the ligand or complexes that have missing fragments on the backbone or sidechain of pocket residues, (7) Exclusion of complexes with ligands containing B, Be, Si and metal elements, (8) Exclusion of complexes where the ligand structure is incomplete, (9) Exclusion of complexes with large ligands exceeding a molecular weight of 1000 or contain 10 or more residues in case of peptides or peptide mimetics, (10) Only affinity data measures as constant of dissociation (Kd) or constant of inhibition (Ki), (11) Exclusion of complexes without precise binding data, (12) Exclusion of affinity data falling outside of the range 2.00–12.00 pKd/pKi, (13) Exclusion of complexes where the protein and/or the ligand in the crystal structure does not match the protein used in the binding assays, (14) Exclusion of complexes where the protein has two or more binding sites and where the bound ligands show more than 10-fold affinity differences.

The BindingDB dataset is a larger collection of binding data holding around 2.7 million binding data points from 1.2 million compounds and 9000 targets. A vast amount of these comes from PDB crystallographic data but they also have docked target series where a set of compounds are docked to the same target with provided experimental binding affinity information.

Binding MOAD contains crystallographic-only poses coming from PDB. It holds 41409 protein-ligand complexes coming from 20387 different ligands and 11058 target families. It is thus a more heterogenous dataset than the PDBBind dataset. However, from the available protein-ligand complexes, only 15223 complexes contain binding data.

A drawback of many binding affinity datasets and consequently ML models is that they represent the protein-ligand binding event statically through only one binding pose whereas binding has both enthalpic and entropic contributions. To overcome this limitation some groups tried to extend datasets like PDBBind with dynamic information. For this, they would run molecular dynamics (MD) simulations for the crystallographic poses in the PDBBind datasets to generate multiple binding poses and simulate the degree of movement of the ligand inside the binding pocket. This way, enthalpic contributions can be more accurately estimated and additional information can be obtained on the entropic contributions of binding. PLAS-5k [39] is one such dataset. Here they selected and simulated 5000 protein-ligand complexes from PDB and calculated several energy components from the MD data such as electrostatic, van der Waals, polar, and non-polar solvation energies. MISATO dataset [40] contains 20000 highly curated protein-ligand complexes from PDBBind. They used semi-empirical quantum mechanics to refine the protonation states of the complexes and fix inconsistencies in the data such as wrong element assignments. They further also computed trace information from MD trajectories as additional information on the degree of flexibility in the binding. This trace information is represented as the degree of movement of each atom across the MD trajectories.

Besides the 3D structural data, other datasets exist comprising a collection of protein-ligand interactions with their respective affinity data without 3D binding poses. While this excludes the use of valuable 3D binding interaction information, it can provide a larger collection of affinity data and focus more on the use of other relationship information for binding affinity prediction, such as the multi-target activity of compounds. The KIBA [50] dataset focuses specifically on kinases comprising 52498 inhibitors against 467 kinase targets. The data was merged from several studies and mapped to CHEMBL and STITCH to enable their comparison and collection of multiple binding affinity information for the same ligand-target interactions. While a direct comparison between IC50 and Kd/Ki scores cannot be performed and conversion between them depends on substrate concentration information (Cheng-Prusoff model [51]: $K_i = IC_{50}/(1 + [S]/K_m)$) that often is lacking in reported binding affinity data, the authors noted on the existence of correlations between IC_{50} and K_d/K_i data in CHEMBL. They used therefore adjustments to the reported K_d and K_i scores based on reported IC_{50} values for the respective protein-ligand interactions. These adjusted K_d and K_i scores were also merged in case both were reported for the same protein-ligand interaction. These adjusted and combined affinity scores, called KIBA scores, span a range describing both binding and non-binding interactions.

3.2.2. *Binding datasets for classification*

All of the previously mentioned binding affinity datasets with the exception of the KIBA dataset provide positive binding information. While this is certainly very valuable information, ML models generally benefit from rich and heterogenous data to learn complex patterns. Therefore, a possible issue of ML models trained on such binding data is their inability to detect non-binding ligands for which one can obtain reasonably good docking poses [52]. This is important since, in practice, virtual screening datasets are comprised of a mixture of potential binders and non-binders. Thus, many ML models for binding affinity are unusable despite showing high performance on binder compounds. In addition, real virtual screening datasets show also a strong unequal distribution in favor of non-binders, which makes the prediction task and selection of the top binding compounds harder. To enable the detection of possible non-binding, decoys molecules specific datasets have been constructed comprising of both positive binders as well as decoys, such as the Dud-e [53], MUV [54] and LIT-PCBA [55] datasets.

The Dud-e dataset [53] is one of the widely used datasets for binding classification. The dataset is a revised and improved version of the preceding Dud dataset [56] that had several internal biases [57–59]. It comprises both 22886 active compounds with activities against 102 targets and 50 generated decoys per active compound. The decoys are generated in such a way that they have similar physicochemical properties as the actives but different 2D topology. Some [60–63], however, have addressed that the Dud-e dataset still has biases between the active and decoy compounds such as the difference in 2D topology, which can be an easy discriminator for ML models to capture. This can lower the practical usability of the trained models. Therefore, better algorithms and datasets have been proposed to overcome these biases, such as the Maximum Unbiased Validation (MUV) or the LIT-PCBA dataset.

The MUV dataset [54] was specifically curated from data taken from PubChem and consists of 15 target subdatasets with 30 active and 15000 decoy molecules each. The data curation consisted of selecting active and inactive compounds confirmed experimentally through both primary and confirmatory bio-activity screens. The actives were further filtered for unwanted compounds such as frequent hitters, high aggregations, or compounds with chromo/fluorogenic properties for screening assays based on optical detection methods. Quality checks on the decoys most similar to the actives were also performed by checking literature sources for any potential binding to the respective target in order to exclude potential false negatives. Lastly, to overcome the structural biases between active and decoy compounds seen in datasets like the Dud-e dataset, the creators of the MUV dataset employed chemical space embedding filters to remove both actives not properly embedded in the decoy chemical subset space and vice versa. This ensures that apart from physicochemical properties the actives and decoys are also structurally similar. Despite the extra effort to reduce bias, it was pointed out [61] that also this dataset has internal bias with decoys not having a proper homogeneous distribution in the chemical space making decoys easy to classify and detect.

Another dataset that was constructed to adjust for the different biases found in the previous datasets is the LIT-PCBA dataset [55]. The dataset contains in total 15 targets with 7844 actives and 407381 inactives. They applied a similar approach as the authors from the MUV dataset, using data from PubChem BioAssays to select confirmed actives and non-active compounds and applying a series of filters to remove non-drug like compounds, compounds with undesired physicochemical properties and compounds that are known to give false positives in many assays such as frequent hitters, compounds with chromo/fluorogenic properties and compounds giving high aggregations in assays. They further selected compounds most similar to compounds found in the PDB database and generated several conformers for each selected compound. The most similar conformer to the PDB ligand was selected for each target set. All compounds were further docked to their respective targets. To overcome the biases found in the previous datasets they used the asymmetric validation method (AVE) [61] to ensure an unbiased selection of actives and inactives in each target sub-dataset. This method measures the pairwise similarities of compounds that belong to one of the 4 subsets (training actives, validation actives, training inactives, validation inactives) and attempts to select training and validation compounds that give the lowest bias scores.

3.3. *Physicochemical and ADMET datasets*

Various datasets exist related to physicochemical molecular properties and ADMET, like PHYSPROP, Tox21, ToxCast, ClinTox and others. The PHYSPROP dataset [64] contains information on 13 physicochemical and environmental fate properties including octanol/water partition coefficients. In total, it contains 47047 chemicals out of which 15806 have octanol/water partition coefficient information. Other datasets on lipophilicity are the logP dataset [65,66] containing 8199 compounds for training together with 3 additional external test sets and the lipophilicity dataset [66,67] which contains 4200 compounds. Several specific datasets for compound solubility have also been curated such as several logS datasets [65,67] with 7799 compounds in total and several external benchmark test sets. The FreeSolv dataset [66,67] which contains solvation free energy data for 642 compounds and the ESOL dataset [67] with information on water solubility for 1128 compounds.

Apart from lipophilicity and aqueous solubility, characterization of compound distribution in the body is also an important aspect that needs to be established. The BBBP dataset [66,67] contains information on blood-brain-barrier penetration for 2039 compounds which is an important property for compounds acting on receptors in the central nervous system and in general for possible toxicity-related aspects. Tox21 [68,69] is a dataset comprising 12707 data points as chemical compounds and results on 12 toxicological endpoints. For each compound in the dataset, 801 dense and 272776 sparse features are included that represent chemical descriptors and chemical substructures respectively. The dataset also comes with training and test subsets making it readily available for machine learning. Another dataset for toxicity information is the ToxCast dataset [70]. This dataset contains around 8000 compounds with results about toxicity on over 600 endpoints. All the data is presented in binary format representing the existence of toxicity against a specific endpoint marker. ClinTox [41] is an interesting dataset that contains toxicity information from successful and failed clinical trials for 1484 drugs. All the negative data was collected from the database for Aggregate Analysis of Clinical Trials (AACT) at ClinicalTrials.gov where only drugs were selected that failed the clinical trial for toxicity reasons. The positive data was selected from DrugBank as FDA-approved drugs. Several toxicity datasets for specific toxicity endpoints are also publicly available. The Quantitative toxicity datasets [66,71] are a collection of 4 datasets with information on LD50 (lethal dose) in rats, IGC50 (50% inhibition growth concentration) of *Tetrahymena pyriformis*, 96-h LC50 (lethal concentration) on *Fathead minnow* and LC50 on *Daphnia magna* with 8307 compounds in total across the 4 datasets and additional external test sets for all 4. The DAT dataset [66] contains information on inhibition and uptake of the dopamine transporter in humans and rats with 887 and 219 class label datapoints on transporter inhibition and uptake information respectively and 1189 and 350 continuous experimental value datapoints on transporter inhibition and uptake information respectively. The hERG dataset [72,73] contains 8 sub-datasets from several literature sources with binary labels of binding and non-binding of the human-ether-a-go-go related potassium channel. In total, the datasets contain 231447 datapoints with additional test sets for each and one dataset containing additional information on the method used to obtain the experimental results. Lastly, the AMES dataset [67] contains 6512 binary classification datapoints related with outcomes for the AMES mutagenicity assay.

4. Molecular properties

4.1. General overview

When looking at the classical pipeline of small molecule drug discovery and development in Fig. 1, we can see at which stages different machine learning based scoring functions can be applied for different properties that need to be established. Additionally, such classical pipelines can also become completely semi-automatic [74]. For this, a combination needs to be made of different computational techniques such as ML models for molecular property predictions, docking software and target and binding pocket identification methods.

Models predicting various physicochemical properties of the selected or generated small molecules are among the first models that are applied. This is to ensure that the molecules selected for binding affinity analysis have already the desired physicochemical properties such as correct protonation state and solubility.

Alternatively one could also screen for binding affinity together with the prediction of the physicochemical properties when using simple models that take only separate protein and ligand information into account (see Section 4.3.2). However, when using models that operate on 3D bound protein-ligand complexes for binding affinity prediction (see Section 4.3.2), it is advisable to employ physicochemical property predictors beforehand to reduce the number of molecules that would go into binding affinity prediction. This is because these models require the molecules to be bound to their target, making docking a bottleneck in the pipeline as ML models in general are capable to produce results instantaneously.

Finally, ADMET models can be applied at a later stage to ensure that optimized leads have favorable absorption, distribution, metabolism, excretion, and toxicity profiles. In principle, these models could also be applied during the initial stages together with the physicochemical predictive models, however, this could potentially reduce the chemical space of compounds tested for binding affinity. One needs to remember that in a pipeline as presented in Fig. 1, experimental testing is performed each time after binding affinity and ADMET predictions of the studied compounds and selection of the top-scoring ones, to experimentally validate the selected compounds. Therefore, using ADMET predictive models early in the pipeline could first of all reduce the diversity of compounds selected for binding affinity estimation leading to a sub-optimal exploration of the chemical space of compounds that can potentially bind well to the target. Second, it would require additional experiments on ADMET profile estimation early on a larger selection of molecules. Taking the higher cost of these experiments into account, this could result in being cost-ineffective. Lastly, one can also observe that two cycles exist in the pipeline for hit and lead optimization. These cycles represent consecutive compound optimization, their screening with predictive models, and experimental validation in order to further improve and select compounds with better molecular properties.

4.2. Physicochemical properties predictive models

Physicochemical properties such as solubility, lipophilicity, and pKa

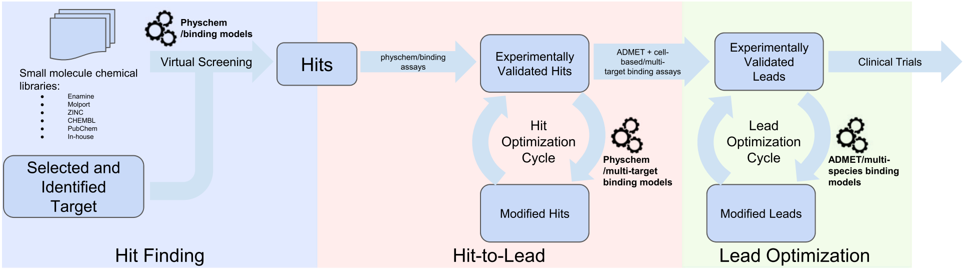
Small molecule chemical libraries:
Enamine
Molport
ZINC
CHEMBL
PubChem
In-house
Physchem /binding models
Virtual Screening
Hits
Selected and Identified Target
Hit Finding
physchem/binding assays
Experimentally Validated Hits
Hit Optimization Cycle
Physchem /multi-target binding models
Modified Hits
Hit-to-Lead
ADMET + cell-based/multi-target binding assays
Experimentally Validated Leads
Clinical Trials
Lead Optimization Cycle
ADMET/multi-species binding models
Modified Leads
Lead Optimization

Figure 1. Workflow of computational hit finding and lead generation. During hit finding, large libraries of small molecule purchasable compounds are screened through predictive models for physicochemical and binding affinity properties against the specific target of interest. These best-scoring hits are further validated through physicochemical and binding assays. Once validated hits are obtained with moderately desired molecular properties, these can further undergo optimization through multiple consecutive rounds of modifications, followed up by virtual screening with predictive models and experimental validation. Important properties to optimize hereby are their physicochemical properties and binding affinity against the main target while ensuring low binding affinity against similar targets to ensure the compound's selectivity. Once optimized hits are obtained for these molecular properties, they undergo more thorough experimental validation using more expensive cell-based and multi-target binding assays and ADMET property assays. The obtained best-scoring lead candidates further undergo a second optimization cycle to improve their ADMET-related properties and ensure target selectivity and second-species validation.

[75] are important properties that need to be established early in the hit and lead generation cycle. As they can influence binding affinity and ADMET properties, it is advisable to establish them early to minimize drug attrition in the subsequent stages. Traditionally these properties have been measured through classical wet-lab experiments such as spectrographic analysis or capillary electrophoretic mobility for pKa determination, plate partitioning, reversed-phase high performance liquid chromatography (HPLC) or capillary electrophoresis for lipophilicity and UV or light scattering for solubility measurements [75]. Various computational techniques have also been developed to predict these properties from empirical algorithms to more advanced machine learning models. Some advancements in the latter are shown in Table 2 and will be discussed further in this section.

4.2.1. pKa predictive models

It is important during early screening and novel compound generation, to assign the molecules their correct protonation state at physiological pH. For this, detection of the protonation sites and accurate pKa prediction for each is important. This should happen prior to the prediction of any other property due to the influence of the molecule's protonation state [92] on any other prediction. Different models have been developed using various ways to embed the chemical information presented to the ML models (Table 2). Common ways to embed the molecules are through chemical and structural descriptors or molecular network graphs.

Chemical and structural descriptors: Molecular chemical and structural descriptors are a type of embedding that constructs a 2D feature vector encoding physicochemical or structural properties of the compounds through different techniques such as rooted topological fingerprints, molecular property embeddings, structural fingerprints or QM-based descriptors.

Lu et al. (2019) [76] uses rooted topological fingerprints to generate embeddings that can be coupled with various ML models. These fingerprints are specifically designed to capture structural information around a central atom. They can do this in 3 ways: (1) through RPairs [93] which embed atom pairs starting from the central root atom and that can be located at n-bonds distance from each other, (2) through RTorsions [94] which is a path-based fingerprint method that constructs the embedding vector from paths starting at the central root atom and a maximum n-bonds distance, (3) RMorgan [76] which is based on the Morgan fingerprint [95], a type of circular fingerprint that embeds information from the central root atom and all the neighboring atoms within a radius of n-bonds. The n distance is each time specified by the user. In all the 3 ways the atoms are defined by a short vector that holds different types of information. In RPairs this vector embeds information about the atomic element, the number of bonded non-hydrogen atoms, and the number of bonding π electrons that it has. In RTorsion the vector embeds information about the atomic element, the number of π electron pairs between the consecutive atoms in the path and the number of non-hydrogen atom neighbors that are not in this path. In RMorgan the vector embeds information about the atomic element, the number of non-hydrogen atom neighbors, the number of attached hydrogen atoms, the atomic charge, isotope information and ring membership.

Table 2
Overview of predictive models for pKa, solubility and lipophilicity prediction. Underlined the best-performing models from each reviewed publications.

Embedding type	Tested models
pKa Prediction Models	
Rooted topological torsion FPs	RF, PLS, XGBoost **Lasso** [76], SVR
Molecular descriptors	**RBF NN** [77], SVM, XGBoost, DNN
Molecular + structural descriptors	**SVM**[78], XGBoost, **DNN** [78], **RF** [79], SVR, MLP, XGBoost
DFT descriptors	PLS, **RBF NN** [80], RF, Gaussian processes, **KRR** [81], feed-forward NN
Molecular graph	**MPNN** [82,83], **AttentiveFP** [84], **GCN** [85]
Aquaeous Solubility Prediction Models	
Molecular + structural descriptors	**XGBoost**[86]
Molecular graph	**GCN** [87], **GAT** [88], MPNN, AttentiveFP
Lipophilicity Prediction Models	
Molecular descriptors	**RF** [89], SVM, XGBoost, **MLP** [90]
Molecular + structural descriptors	RF, XGBoost, MLR, **FFNN**[91]
Molecular graph	GCN, **GAT**[88], MPNN, AttentiveFP

Abbreviations: GNN = Graph Neural Network, RF = Random Forest, PLS = Partial Least Squares, XGBoost = Extreme gradient Boosted Trees, SVM = Support Vector Machine, KRR = Kernel Ridge Regression, NN = Neural Network, RBF = Radial Basis Function, kNN = k-Nearest Neighbours, DNN = Deep Neural Network, GCN = Graph Convolutional Neural Network, GAT = Graph Attention Network, MPNN = Message Passing Neural Network, FP = Fingerprint, MLR = Multivariate Linear Regression, FFNN = feed forward neural network

In Lu et al. (2019) [76] they combined the constructed fingerprints with random forest, XGBoost, PLS, Lasso regression, SVM or kNN models. They saw that the RTorsion embedding method provided the best results coupled with Lasso regression. XGBoost and SVM showed also close to top performance on their time-split-based test set. The embedding described here is a structural descriptor type embedding as it tries to use information on the structural arrangement of the molecule.

Another group of descriptors that can be used are chemical descriptors which provide a series of calculated molecular properties such as molecular weight, logP/logD, quantitative estimate of drug-likeness (QED) [96], number of hydrogen bond donors or acceptors and many others. All these can be easily computed using the RDKit [97] Python library. In Baltruschat et al. (2020) [79] they used 196 molecular descriptors together with a 4096-bit long structural Morgan fingerprint [95] with radius 3 as embedding vector coupled with random forests, SVMs, MLP and XGBoost models. Morgan fingerprints coupled with random forests showed the best performance on two test sets, the publicly available Novartis test set [98] and the own curated set from several literature sources [99–102,103]. While they report good accuracies on the literature test set, this method has the disadvantage that pKa values for each specific protonation site of the molecule cannot be predicted, as the whole molecule is embedded into a single vector. Therefore this method cannot be used for structures bearing multiple protonation sites like amphoteric molecules.

Mansouri et al. (2019) [78] use also a combination of molecular descriptors, binary structural fingerprints, and fragment counts. The latter splits the molecule into smaller fragments and generates a fingerprint based on the number and type of fragments in the whole molecular structure. They coupled this embedding with SVMs and kNN models. While their approach also does not make it possible to generate predictions for individual protonation sites, they solve the problem of amphoteric molecules by applying a step-wise prediction approach using both classification and regression models. They first build classification models that classify the molecules into one of three groups: (1) acidic molecule, (2) basic molecule, and (3) amphoteric molecule. The acidic and basic groups are established in the same way as in MolGpka [83]. For each class, they construct separate regression models predicting the basic and acidic pKa separately. While such an approach allows to generate predictions for amphoteric molecules containing a basic and acidic protonation site, they still cannot be used for molecules with more complex protonation profiles, containing multiple protonation sites. As seen in Lu et al. (2019) [76], in order to be able to generate predictions for multi-protic compounds, it is important to generate local embeddings that take information primarily from the local neighborhood of the protonation site. Where in Lu et al. (2019) [76] this was achieved through structure-based fingerprints, Hunt et al. (2020) [80] use QM chemical descriptors to embed each protonation site of the input molecule. The used descriptors capture the atomic and bond properties of the atom bearing the protonation site with its surrounding bonded hydrogen and heavy-atom neighbors. Properties such as nucleophilic and electrophilic delocalizabilities, bond lengths and atom charges were used together with information on the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO) energies and heats of formation. For polyprotic compounds, most acidic and basic sites were first established following a step-wise approach to locate other protonation sites while maintaining the correct protonation for already established sites. This allows for a more accurate calculation of the QM descriptors. They further tested the generated embedding with linear, partial least squares (PLS), radial basis functions (RBF), random forest (RF) and Gaussian process (GP) ML models where RBF, GP and RF models showed the best performance.

In Lawler et al. (2021) [81] both protonation-centric and whole molecule chemical descriptors are used to construct the molecular embedding such as information on the electronegativity of the central atom, magnitude of the dipole moment of the whole molecule, the degree of oxidation of the central atom, number of hydrogens in the whole molecule, number of fluorine and carbon atoms which say something about major electron-withdrawing groups and size of the overall molecule respectively, the molecule's molecular weight, the Connolly volume [104] of the molecule, solvation free energy and DFT calculated pKa values for the specific protonation site. By using a DFT-computed pKa value as a molecular descriptor, this method could be seen as a further refinement of the DFT-calculated pKa values by taking other types of chemical information into account. These molecular embeddings are further coupled with ML models such as kernel ridge regression (KRR), Gaussian processes and feed-forward neural networks. Hereby KRR gave the best performance and reported a lower error with the experimental pKa values than the initial DFT-calculated pKa values.

Molecular network graph embedding: Graph-based models use node and edge chemical feature embeddings with a GNN. The input is a small molecule that is transformed into a network graph with nodes and edges as atoms and bonds respectively. Each node and edge is further embedded through 2D vectors that can be constructed according to different rules.

MolGpka [83] uses 1-hot embedding for the nodes consisting of a total of 39 bits that encode information such as the atom element, its hybridization state, whether the atom is a hydrogen bond donor or acceptor, the atom's degree, its valence, whether it forms part of a cyclic structure and the size of the ring, whether the atom is aromatic and whether it is the protonation site. For the edges, they use a binary adjacency matrix representing the connectivity of the nodes according to existing chemical bonds.

Graph-pKa [84] uses also 1-hot encoded node embeddings similar to MolGpka [83]. The difference between their node embeddings lies in a different number of bits for atom element information, the atom's degree and the hybridization state of the atom. Further MolGpka [83] encodes information about hydrogen donating and accepting properties of the atom and whether the atom is the protonation site which is not used in Graph-pKa [84] node embedding. While the latter embeds information on the formal charge, presence of radical electrons, number of bonded hydrogens and chirality. Different from MolGpka [83] they also apply a 1-hot encoded embedding vector on the edges which are taken to be chemical bonds between the atoms. The edge embedding vector encodes information about the type of bond, whether it is conjugated or not, whether it is part of a cyclic structure and stereo-chemical information.

Similar to this, MF-SuP-pKa [82] uses a 40-bit long node embedding vector based on Xiong et al. (2020) [105] adding 1 additional bit for the atomic degree information. They also apply a 1-hot encoded embedding vector for the edges based again on the one used in Xiong et al. (2020) [105] and add 2 additional bits for the stereo-chemistry information.

Epik [85] also uses a node embedding vector based on chemical information. They use a 74-bit long vector encoding information about the atomic element, its degree, the number of valence and radical electrons, the formal charge, the hybridization state of the atom, whether it is aromatic and the number of explicit bonded hydrogens. The main difference between their embedding vector and the previous ones is that they encoded a larger number of possible atomic elements, use more bits to represent the atom's degree and use a mixture of 1-hot encoded and single continuous values in the embedding. Similar as in MolGpka [83] they establish edges based on bond connectivity of the atoms.

Once nodes and edges are embedded, information is exchanged between the nodes through either message-passing layers like in Graph-pKa [84] and MF-SuP-pKa [82] or graph convolutions like in MolGpka [83] and Epik [85]. Both Graph-pKa [84] and MF-SuP-pKa [82] use the attention-based message passing architecture from Xiong et al. (2020) [105]. This uses a node-level attention vector constructed from a weighted combination of the node's neighbors which is merged with the node's updated embedding vector at each layer through gated recurrent units (GRUs) which help to take influences of further away located nodes into account. Hereby, edge embedding vector information is first added to the node embedding vectors each time prior to the attention-based message passing operation.

While Graph-pKa [84] uses this type of message passing on the complete molecular graph centered around the atom bearing the protonation site, MF-SuP-pKa [82] defines first k-hop molecular substructures around each atom that holds the protonation site and applies an intermediate weighted pooling operation within the nodes of each molecular substructure followed by additional attention-based message passing operations on the pooled super-nodes of each molecular substructure. Different in MolGpka [83] from Epik [85] is that in the prior, two separate networks are used for acidic and basic protonation sites respectively. This is done in order to keep the input molecule in its neutral state while in Epik [85] the input molecule's protonation site(s) is/are always in the protonated form with respect to their experimental pKa value. Finally, all models use fully connected layers to reduce the embedding vector of the node that holds the protonation center to the scalar prediction value for the pKa.

Traditionally, neural network models are trained by back-propagating the gradient of the loss to each parameter in the network. CSAPSO-EDCD RBF ANN [77] uses an optimized particle swarm optimization (PSO) algorithm [106,107] to select the input features and train their neural network. PSO is a type of genetic algorithm that tries to optimize a set of parameters in parallel through a search for the best parameter combination taking into account the value of the other parameters. Different from the other methods, they do not model the molecules as graphs but instead, compute a series of 686 molecular descriptors that are further reduced to 5 using their improved PSO algorithm. These were then coupled with a radial basis function (RBF) neural network, which is a type of feed-forward neural network typically characterized by an input layer, 1 hidden layer that uses radial basis functions as activation functions and 1 linear output layer. In this work, PSO was further applied to find the variables of the RBF functions of each neuron that fit the data. Different from the previous methods, this method does not embed each protonation site inside the molecule but instead generates a single whole molecule embedding and would therefore fail on multi-protic compounds.

Performance comparison: In order to compare the performance of models it is important to use standard benchmark test sets to make the comparison as least biased as possible. However, benchmark test sets used in the different works on pKa prediction show different test sets used to benchmark their models. This makes direct comparisons between them hard. In general, all methods report high performance with squared Pearson's correlations above 0.90. Baltruschat et al. (2020) [79] does report lower performance on the Novartis test set [98], probably because of the higher heterogeneity and size of the compounds in the test set. This indicates that some of the models have certain limitations as discussed above in terms of generating predictions for more complex multi-protic compounds. Therefore, methods that model each protonation site separately can be more universally applied to different small molecules as opposed to methods that generate a single embedding for the whole molecule. Interestingly, despite being more simple, non-neural network methods are able to also achieve competitive performance as more complex graph-based neural network models.

4.2.2. Aqueous solubility predictive models

After adjusting the molecules to the correct protonation states one of the other important physicochemical properties to predict is the aqueous solubility, logS (in mol/L). As this property is generally an effect of the whole molecular structure rather than local molecular neighborhoods like in pKa prediction, one can easily get good performance by applying whole molecule chemical and structural embeddings coupled with simple linear or decision tree-based models.

Falcòn-Cano et al. (2020a) [86] uses 1400 whole molecule chemical descriptors together with 45 additional physicochemical properties. To reduce the number of features in the generated embeddings they applied a selection by permutation of the variables using a random forest model where only high occurring variables were selected in the individual decision trees of the random forest model, together with recursively selecting the most correlated variables. This was further coupled with both classifier and regression XGBoost models as solubility can span a wide range of values. To do this, training data was classified into a soluble and highly soluble class (logS ≥ -2) and a slightly soluble and insoluble class (logS < -2) to be used to train the classifier model. Separate regression models were further trained for the two separate classes. Additionally, a third regression model was trained on all the training data and an ensemble approach was used by taking the average of the local and global regression models. The method was tested against two external test sets with curated data from the literature. The performance of the final regression model showed a median performance of around 0.64 and 0.69 Pearson's correlation for models trained on cleaned data points only and extended with reliable data points respectively based on the accuracy of the reported experimental logS values on the test set 1 and 0.43 for test set 2. The performance of the classifier model showed a good performance of 0.80 accuracy, 0.60 Cohen's Kappa, 0.89 sensitivity and 0.71 specificity for test set 1 and 0.83 accuracy, 0.67 Cohen's Kappa, 0.73 sensitivity and 0.93 specificity for test set 2 when using only cleaned data points.

Different from this, Chemi-Net [87] uses a graph-based neural network, similarly as described in 4.2.1. They used both atom and bond chemical descriptor-based embedding vectors containing information about the atom type, van der Waals and covalent radius of the atom, the number of rings the atom belongs to, whether the atom is in an aromatic ring or not, and the electrostatic charge of the atom for atom embeddings. For bond embeddings they used the bond type, bond length, and whether the bond is part of a ring system. They used a convolutional graph neural network to update the atom and bond embeddings and applied several pooling and dimensionality-reducing layers to generate the output value. Interestingly, they trained their model on several molecular properties in parallel such as aqueous solubility, CYP450 inhibition, human liver microsomes, bioavailability, and PXR induction. All these other properties are ADMET specific and will be discussed further in Section 4.4. To train the model in such a parallel fashion, they applied a combined loss function over the individual predictions. This allows the model to learn across datasets and improve performance on each sub-dataset especially when few data points are available. The performance varies depending on the molecular property predicted with Pearson's correlations ranging between 0.11 and 0.692 when the model is trained on one single task only. When trained with the multi-task loss, performance improves for some of the properties, including the low-performing ones for which performance increased for example from 0.2 to 0.327 Pearson's correlation. This shows that multi-task learning can help when a few data points are available for one task. Also, performance seems to correlate only slightly with training data size. Computing Pearson's correlation between the obtained performances on the test sets for the single-task learning and the training set sizes gave a correlation of 0.136. A similar correlation was obtained using test set sizes. This indicates that amount of training data has importance to some degree but that data quality is equally if not more important.

4.2.3. Lipophilicity predictive models

A third important physicochemical property that needs to be established is lipophilicity as it has important implications for the molecule's solubility and membrane passage. This is expressed as partition coefficients of the compounds between a hydrophobic and a hydrophilic environment either as logP values for non-ionizable compounds or logD values for ionizable compounds where the distribution of the compound between the two phases depends on the fraction of the ionized and non-ionized species which is influenced by the surrounding pH. Again, just as in aqueous solubility, the property depends on the whole molecule. Therefore, whole molecule chemical and/or structural descriptors can be generated and coupled with different ML methods.

Win et al. (2023) [90] used 204 chemical descriptors from RDKit [97] which were further pruned to 125 excluding descriptors with low variance, high correlation to other descriptors and descriptors with missing and zero values. These chemical descriptors were further augmented with structural Morgan fingerprints [95] and experimental reversed phase chromatography retention times. These embeddings were further tested with several ML methods such as SVM, MLP, XGBoost and random forests. They constructed separate models to predict both logP and logD values. The MLP model gave the best performance on the validation set and gave high performance on the test set with Pearson's correlation values above 0.85 on both metrics. From further feature interpretability they found unsurprisingly that the retention time had the most impact on the predictions.

Just as in Chemi-Net [87], in Wenzel et al. (2019) [91] they used multi-task learning to train a deep neural network using chemical molecular descriptors with atom-pair and pharmacophoric donor-acceptor pair descriptors. They train the model on a set of different molecular properties such as metabolic clearance, passive permeability in Caco-2 cells, metabolic liability, and logD values. Different from Chemi-Net [87], they trained their model in a step-wise manner by training the model on one task, optimizing the shared weights and task-specific weights of the neural network model and keeping weights for other tasks frozen. This makes it possible to train the network by using different independent datasets with a limited overlap of compounds. From the results, they show that by combining related datasets where the properties have a certain relationship such as data on metabolic liability in different animal species, performance improves compared to a single-task trained model. However, combining unrelated datasets such as metabolic liability data with logD does not always give an improvement in performance. In general, performances range from around 0.65 squared Pearson's correlation for metabolic liability datasets to around 0.85 for logD prediction with data from in-house company experimental screening showing more consistency than when using publicly available data from CHEMBL, where performances could drop to around 0.50 on some benchmarks.

In Broccatelli et al. (2022) [88] again they trained graph-based models in a multi-task approach using data for different molecular properties such as logD, intrinsic clearance in human liver microsomes and hepatocytes, and kinetic solubility in a phosphate buffer. They apply 1-hot encoded chemical-based atom and bond embeddings similar to some previously seen graph-based models with information on the atom type and its degree, whether the atom is chiral, its formal charge, hybridization state, number of implicit valences, whether the atom is aromatic, the bond type, whether the bond is conjugated and part of a ring system and stereo configuration of the bond. They used different types of graph-based models such as graph convolutional networks, graph attention networks, message passing networks, and the attentive fingerprint model from Xiong et al. (2020) [105]. For the multi-task prediction, they tested two approaches: (1) using task-shared and task-specific layers in the neural network models such as in Chemi-Net [87] and (2) a bypass architecture where separate neural networks are trained for each single task and a general model trained for all tasks. The final prediction is then the ensemble of the output of both the task-specific and general models. The graph attention network showed better performance in single-task learning but multi-task learning did not show always an improvement over single-task learning. It had small improvements for prediction of molecular properties such as solubility, metabolic liability and clearance which can benefit from information such as lipophilicity, but not vice versa. In general, performances ranged from 0.30 to 0.63 squared Pearson's correlation for the different properties evaluated on time-split-based test sets indicating average performance.

4.3. Binding affinity predictive models

The next important property that needs estimation is the binding affinity of a complex. The binding prediction can be performed in three different ways: (1) as a classification where the model classifies compounds as binders or non-binders or into different binding affinity ranges; (2) absolute, where the model predicts the binding affinity metric directly of the molecule to its target; (3) as a prediction of the relative binding affinity between pairs of compounds binding the same target. Further, the three ways will be discussed together with their existing models.

4.3.1. Binding affinity classifiers

Classifier models perform classification of the input bound complexes into classes based on binding/non-binding of the ligand or a specific binding affinity range. When looking at the type of machine learning models that exist for absolute binding affinity prediction, we can find a large range of diverse models that use similar model types as in physicochemical properties prediction but that embed the input data through different methods (Table 3). This embedding can be performed by either using ligand-only input for single target datasets, separate protein and ligand representations, or by taking the bound protein-ligand complex and using binding energies, interaction fingerprints, or 3D representations to construct embeddings. While some other embedding and model types can be found in prediction models for absolute binding affinity, theoretically any embedding and machine learning method used for absolute binding affinity prediction can also be used for classification, with a clear example being BindScope [108] which is very similar to KDeep [109]. The main difference lies only in the final output of the models where in the case of classifiers the output represents probabilities of binding or class-specific probabilities in the case of multi-class classification.

In Morris et al. (2020) [110] they use only ligand information in the form of embeddings generated by a text-based transformer neural network that was pre-trained on a large number of small molecules through a translation pretraining task to predict the molecules' IUPAC names from their SMILES strings. The intermediate latent embeddings were further used with a feed-forward neural network trained to classify binders and non-binders in different single-target datasets. They showed improved performance from using such latent embeddings obtained through a pre-trained transformer network as opposed to embeddings generated by a non-pretrained model. While they report good performance across different targets, the important drawbacks of such a model are first that it can only operate on single targets, meaning that a sufficient amount of data needs to be available to train the classifier models. Second, such a method could fail to classify correctly binders and non-binders that are similar in both structural and physicochemical properties as the model does not obtain any additional target-related information.

Torng et al. (2019) [111] on the other hand uses separate protein and ligand representations. They do this by using graph representations of the target's binding pocket and ligand's molecular structure. These graph representations consist of 2D network graphs of either the protein binding pocket or the ligand structure. For the protein representation, nodes and edges represent residues and connections between

Table 3
Overview of ML methods for binding affinity predictions of different types. Underlined are the best-performing models from the tested ones. References with an asterisk use separate protein and ligand representations as input instead of protein-ligand binding complexes and references with a double asterisk use ligand-only representations.

Method name	Embedding type	Tested models
Binding Affinity Classification		
Morris et al. (2020) [110]* *	Text-based transformer embedding	**FFNN**
Torng et al. (2019) [111]*	Protein pocket & ligand graph embedding	**GCN**
vScreenML [112]	Rosetta energy terms	**XGBoost**
Nogueira et al. (2019) [113]	PADIF interaction FPs	FFNN, **SVM**
BindScope [108]	3D voxels	**3D CNN**
Lim et al. (2019) [114]	Molecular graph embedding	**GAT**
Absolute Binding Affinity		
ChemBoost[115]*	Ligand SMILES embedding + ligand-based protein embedding	**XGBoost**
DeepFusionDTA [116]*	Ligand SMILES embedding + protein sequence embedding	**light GBM**
AttentionDTA [117]*	Ligand SMILES embedding + protein sequence embedding	**1D CNNwith attention**
DeepDTA [118]*	Ligand SMILES embedding + protein sequence embedding	**1D CNN**
SimCNN-DTA [119]*	Ligand-ligand and protein-protein similarities	**2D CNN**
ECIF-LD-GBT [120]	ECIF + ligand chemical descriptors	**XGBoost**
Wang et al. (2021a) [121]	Proteo-chemometrics IFP	**RF**, GBDT, FFNN, DT
BAPA [122]	Interaction fingerprint + Vina energy terms	**1D CNNwith attention**
ET-score [123]	Distance weighted interaction fingerprint	**ERT**
SMPLIP-Score [124]	Interaction fingerprint + ligand fragment embeddings	**RF**, DNN
Taba [125]	Mass-spring distance weighted interaction fingerprints	LR, LAS, lasso, RR, **elastic net**^{a}
Zhu et al. (2020) [126]	Protein-ligand pairwise interactions	**FFNN**
OnionNet [127]	Shell-established protein-ligand interaction atom pair counts	**2D CNN**
Wojcikowski et al. (2018) [128]	PLEC fingerprint	LR, **RF**, NN
Leidner et al. (2019) [129]	Protein residue centered interaction FPs	**XGBoost**
PotentialNet [130]	Adjacency-based atomic interactions	**2D CNN**
3D-RISM-AI [131]	Hydration free energy properties + SASA + rotatable bonds	RR, SVM, RF, **XGBoost**
Δ_{vina}XGB [132]	Vina energy terms	**XGBoost**
GXLE [133]	Molecular mechanics energy terms + physical interaction energy + empirical interaction energy + ligand descriptors	LR, RR, DT, **XGBoost**, SVM, RF, DNN
Boyles et al. (2019) [134]	Structure-based energy descriptors of protein-ligand complex + ligand chemical descriptors	**RF**
Fujimoto et al. (2022) [135]	PMF + ligand MACCS and ECFP + custom protein AA count vectors	Lasso, light GBM
RASPD+ [136]	Protein/ligand chemical descriptors	RF, SVR, DNN, LR, kNN
PerSpectML [137], FPRC-GBT [138]	Spectral graph properties	**XGBoost**

(continued on next page)

Table 3 (*continued*)

Method name	Embedding type	Tested models
Nguyen et al. (2018) [139]	Spectral graph properties	RF, 1D CNN, **ensemble**
AGL-Score [140]	Spectral graph properties	**GBT**
PPS-ML [141]	Path spectral features	**GBT**
^{SYBYL}GGL-Score [142]	Graph colouring using protein atom names and SYBYL ligand atom types	**^{SYBYL}GGL-Score**, ^{ecif}GGL-Score
KDeep [109], DeepAtom [143] Pafnucy [144], Francoeur et al. (2020) [145], AK-Score [146]	3D voxels	**3D CNN**
AEScore [147]	Atomic environment vector	**ANI NN**
GAT-Score [148]	Atom and bond feature vectors	**GAT**
ECIFGraph::HM-Holo-Apo [149]	Protein-water & protein-ligand-water graph representations	**Graph transformer**
Relative Binding Affinity		
DeltaDelta [150]	3D Voxels	**2-leg 3D CNN**
Gusev et al. (2023) [151]	Path-based FPs, Morgan FP, 3D molecular FP, PLEC FP and combination of 3D and PLEC FPs	RF, MLP, LR, kNN, SVM, GP, GP with Tanimoto kernel

Abbreviations: SASA = solvent accessible surface area, AA = amino acid, RR = ridge regression, SVM = support vector machine, RF = random forest, CNN = convolutional neural network, XGBoost = extreme gradient boosting, MACCS = molecular access systems key fingerprint, ECFP = extended connectivity fingerprint, PMF = potential of mean force, ECIF = extended connectivity interaction features, FP = fingerprint, IFP = interaction fingerprint, GBDT = gradient boosted decision trees, GAT = graph attention network, DNN = deep neural network, GBM = gradient boosted model, PLEC = protein-ligand extended connectivity, GBT = gradient boosted trees, PADIF = protein atom score contributions derived interaction fingerprint, MLP = multilayer perceptron, LR = linear regression, DT = decision tree, SVM = support vector machine, SVR = support vector regression, FFNN = feed-forward neural network, ERT = extremely randomized trees, LAS = least absolute shrinkage, GP = gaussian processes. [a]elastic net combines lasso and ridge parametrization in linear regression

neighboring residues respectively, while for the ligand representation, they represent the atoms and molecular bonds. Both representations pass through a GCN network and both learned latent protein pocket and ligand embeddings are concatenated before a class probability is returned in the final output layer. They further also use two encoder architectures for the protein pocket embedding which first learn latent embeddings for each protein pocket residue through neighboring residues followed by a mapping of these residue latent embeddings into a 2D feature vector. Interestingly, in this work, the use of pretraining of the protein pocket embedding layers by using an auto-encoder setup learns to recover the protein pocket network graph in an unsupervised manner. This is done as usual training sets for binding affinity classification have a limited amount of diverse targets.

Differently from the previous models, all other models and methods use representations of the bound protein-ligand complex. In vScreenML [112] they used Rosetta energy terms coming from the Rosetta model [152] for energy prediction of biological systems. These were then further coupled to an XGBoost ML model to learn non-linear relationships between the different energy terms and the classification of targets into binders and non-binders. They especially took additional care to prepare debiased training sets by selecting decoys using the Dud-e server [53] to select decoys that would match physicochemical properties with the binders but have different structural arrangements. Then low energy docking poses were generated for each selected decoy and these were mapped to minimized crystal poses of actives to ensure a good overlap of general shape and charge distribution. While this approach would enable the selection of decoys that match the physicochemical and overall structure to the binders, it could still potentially contain bias firstly, due to the use of Dud-e, which as reported [60,62,63] and discussed in Section 3 has its own intrinsic bias, and secondly, due to the fact that initially selected decoys are structurally not completely similar to the active molecules. It could therefore be better to use more debiased datasets such as MUV [54] or LIT-PCBA [55] as discussed in Section 3.

In Nogueira et al. (2019) [113] they used protein per atom score contributions derived interaction fingerprints (PADIF) which embed the interaction patterns between ligands and their targets. They tested these embeddings with both feed-forward neural networks and SVMs and noticed that SVM had a slightly better performance across different test sets. Interestingly in this work is that they used experimentally verified decoy molecules from assay data in the CHEMBL dataset [43]. This helps to reduce the risk of selecting false negative decoys. Further, they also performed additional tests on inter-target selectivity where active compounds were cross-docked with other target families with assigned decoy labels for those. While this could add the risk of false negatives, statistically the chance of this happening would still be low. They found that their model had sufficient sensitivity to detect differences in the change of target which means that the model was able to learn specific protein-ligand interaction terms.

In BindScope [108] a similar model is used as in KDeep [109] for absolute binding affinity prediction. Both models employ a 3D voxelized representation of the protein-ligand bound complex. Here, a 3D representation of the protein-ligand binding complex gets broken up into voxels, which are volumetric counterparts of pixels in three dimensions. Just as different RGB channels in colored images, one can define multiple channels in the fourth dimension of these voxelized representations where each channel stores chemical-based information on each voxel. Such information can be aromaticity information, occupancy, hydrogen bond donor and acceptor properties, or electrostatic interaction properties. These properties and their ranges of influence are defined by each atom and its van der Waals radius. These 4D representations are then fed to a 3D CNN architecture that learns to extract hidden features and map them to a binary probability value that represents whether the ligand is a binder or not.

Another way how 3D molecular embeddings can be introduced is through molecular graph embeddings. In Lim et al. (2019) [114] they construct 3D molecular network graphs from 3D representations of the bound complex. Hereby protein-ligand complexes are modeled as 3D graphs where each node represents atoms and edges represent any type of inter-atomic interaction, either bonded or non-bonded. Each node and edge can hereby be defined through physicochemical or quantum mechanical information vectors. These representations can then be fed to a graph-based ML model which learns to map the 3D graph-based structural information to binding affinities. In Lim et al. (2019) [114] they define only the nodes by using information on the atom's element, degree, number of attached hydrogens, valence electrons, and whether it is aromatic. Further, the atom connectivity and inter-atomic distance information is added to model both bonded and non-bonded interatomic interactions and subtractions between both adjacency matrices are performed so that the model can learn differences between both interaction types.

4.3.2. *Absolute binding affinity prediction*

In absolute binding prediction, the binding affinity of a small molecule to its target is predicted as a single value. The binding affinity can hereby be expressed with different metrics like the dissociation constant Kd, the inhibition constant Ki, the inhibition IC_{50} or response concentrations EC_{50} of a target at half-maximal concentration or as a more physics-based metric like the difference in Gibbs free energy ΔG between the bound and unbound state of a target. Both Kd and Ki metrics as well as the ΔG are complex intrinsic measures of binding affinity, meaning that they depend only on the target, ligand, and the interactions that both form. Thermodynamically, they are a result of changes in enthalpic and entropic contributions upon binding according to the below formula.

$$\Delta G = \Delta H - T\Delta S$$

As binding affinity is an equilibrium process between the unbound and bound states of the ligand to its target, the Kd and Ki constants represent ratios between the Kon and Koff reaction constants which in their turn are ratios between the concentrations of free target and ligand and the bound complex of the two. One can convert ΔG values to Kd/Ki values via $\Delta G = RTlnKd$. IC_{50} and EC_{50} metrics on the other hand depend on the concentrations used by the target and the ligand during the assays. Therefore these depend strongly on the experimental conditions and can produce different results for the same ligands and targets when assay conditions change. Therefore, the use of these metrics in ML should be done with care ensuring similar assay conditions when obtaining the experimental binding affinity values. Failure to do so will introduce bias into the models, harming their accuracy.

Again, diverse embedding and machine learning methods can be found. Roughly we can separate them into two main groups which were also seen previously in Section 4.3.1: (1) models that use protein and ligand information where each is presented as separate structures without the explicit information of the binding complex that they form, (2) models that use protein-ligand complex conformation information. In the next sections, a more detailed overview will be given of the models in each group.

Separate protein and ligand information: As high-quality crystallographic binding poses between a ligand and its target are expensive to obtain and docking poses rely heavily on the performance of the docking software, some models try to learn binding information from separate protein and ligand structures. In this group of models, input structures are provided as separate protein and ligand representations without any crystal or docked ligand poses into the protein's binding pocket. This allows them to learn on a larger amount of data for which high-quality binding poses are currently non-existent.

From the models studied in this review, this is mostly done through the embedding of the ligand's SMILES representation and the protein's representation which can be generated either as a concatenation of individual ligand's embeddings for the same protein target [115] or through the embedding of the protein's sequence string [116–118]. These embeddings are usually generated through natural language processing (NLP) models which learn to embed the string representations of ligands and proteins into a 2D vector embedding.

This embedding is further concatenated and used in various ML models ranging from simple boosted trees algorithms like XGBoost [115] and light gradient boosted trees [116] models to more advanced CNN models [118] which can additionally use attention weights [117] or long-short term memory (LSTM) blocks [116] to better learn long-distance relationships in these merged embedding vectors.

Alternatively, protein and ligand information can also be presented through pairwise similarity matrices that represent how similar each protein and ligand are to other proteins and ligands in the training set. Hereby, one can employ classic distance measures like the Tanimoto similarity on 2D ligand embeddings through any embedding method such as Morgan fingerprints [95] and protein sequence similarities as computed through programs like the Basic Local Alignment Search Tool (BLAST) [153]. Such representations use the idea that similar ligands bind to similar types of protein targets and therefore would also exhibit similar binding affinity properties. These representations can then be fed to 2D CNN models that are able to extract hidden relationships between the similar ligand and target information to estimate binding affinity for new, unknown ligands or targets.

Protein-ligand complex information: While models using information from separate protein and ligand structures have shown to have a decent performance, they omit important structural information from the 3D binding complex between a ligand and its targets. Therefore, a large number of models exist that try to learn explicitly from the binding information. To do this, binding complexes are provided either as crystallographic or docked poses and embedded via various methods such as interaction fingerprints, molecular descriptors of binding, spectral graph properties, or voxel-based or graph-based representations of the binding complex.

A largely used embedding method is the use of interaction fingerprints. These fingerprints, just as their molecular structural embedding counterparts such as Morgan [95], the Extended Connectivity Fingerprint (ECFP) [154] or MACCS [155] fingerprints used for small molecule compounds, embed specifically the interaction patterns between ligand compounds and their targets. Various such fingerprints like the Extended Connectivity Interaction Features (ECIF) [120] and the Protein-Ligand Extended Connectivity (PLEC) [128] use similar methodologies as in the classical Morgan [95], ECFP [154] or MACCS [155] fingerprints to embed information between neighboring ligand and protein groups. Hereby, ligand and protein-specific information can be represented with different types of information. For the ligand, this can be atomic element information, explicit valence information, number of bonded heavy atoms, number of bonded hydrogens, aromaticity, and ring membership. For the protein, this can be atomic element and residue information. The distance between the groups can be defined in several ways. One can incorporate the distance between central atoms of the functional groups in the ligand and protein side into the vector [123, 128], whereby the distance is rigid, or dynamic, modeled by mass-spring-like functions [125]. One can also generate multiple distance shell radii [127] and embed interaction information for each such shell individually and concatenate all this information to obtain interaction embedded information across multiple distances for each ligand functional group or atom.

Usually, such interaction fingerprints are ligand centric, meaning that they start from the ligand atoms or functional groups and embed both ligand and protein information in their direct neighborhood. Alternatively, one can also construct them protein-centric [129] whereby one would embed close ligand information around protein residues located in the binding pocket to obtain concatenated interaction information for the different protein residues. Here they would use features such as contact van der Waals potential between residue-ligand, protein-ligand hydrogen bonds, protein-ligand halogen bonds, protein-ligand salt bridges, π-interactions, and π-cation interactions.

Such fingerprints also embed information that is in immediate proximity to the functional groups, as local protein-ligand interactions are what is driving the binding between both. One can add additional further distance information, as mentioned before, through shells [127] or by using all possible pairwise protein-ligand interactions [126]. This latter might prove to be computationally more expensive, especially with the growing size of the binding complex, and therefore more coarse-grained representations could prove to be useful there to reduce the number of pairs. In Zhu et al. (2020) [126] they use distance information to reduce the number of pairs and add additional quantum mechanical energy terms to the featurization such as partial charges and Lennard-Jones parameters.

These interaction fingerprints also often use molecular structural information in the form of atomic elements, functional groups, and their bonded and non-bonded interactions. These can be further expanded to include other atomic information like formal charges, hybridization states or ring information, proteo-chemometric information [121], quantum mechanical energy terms [122], or ligand specific information as either chemical descriptors [120] or fragment embeddings [124]. These latter, proved [120] to be useful to further improve the performance of the models and in Boyles et al. (2019) [134] they also found that training on ligand information alone would teach the model an average binding affinity score for that ligand across its different protein targets.

All these different interaction fingerprints are constructed as flat 2D vectors that can be fed to a wide array of ML models like linear models with or without regularizations like lasso, ridge or least absolute shrinkage, decision trees, random forests, gradient boosted trees and forests and neural networks like feed-forward neural networks or CNNs with or without additional attention mechanisms. Alternatively, these interaction fingerprints can also be based on adjacency matrices as in PotentialNet [130] taking into account adjacency information between ligand and protein atoms with additional atomic chemical and quantum mechanical descriptors. Such representation can further be fed into 2D CNN architectures to learn hidden features. Wang et al. (2021b) [156] provides further an overview of different possible interaction fingerprints that are used for binding affinity predictions, many of which were also found in the discussed papers. Yin et al. (2023) [157] provides also an interesting study on how different hyperparameters that guide interaction fingerprints construction affect the performance of binding affinity prediction models.

Alternatively one can also construct flat 2D embedding vectors using quantum mechanical descriptors of protein-ligand binding interaction [131,135,133,134,132]. Common descriptors hereby are: hydration free energy properties, solvent-accessible surface area, information on rotatable bonds, physical and empirical interaction energies, or Vina energy terms. The Vina terms consist of protein-ligand interaction terms, ligand property counts, and buried solvent-accessible surface area features. In GXLE [133] they noticed that combining different energy terms usually gives better performance, especially when information embedded in these terms is highly complementary. However, in Nguyen et al. (2018) [139] they noticed that this is not always the case as they found out with the inclusion of additional Vina energy terms. These descriptor embeddings can additionally also be extended with structural ligand embeddings [135] using traditional embedding algorithms such as MACCS [155] or ECFP [154], chemical ligand descriptors [133,134] or protein embedding information like amino acid count vectors [135]. These types of embeddings can further be fed into again a wide array of ML models like linear models, decision tree-based models, or neural networks.

Interactions can also be modeled through calculated chemical molecular descriptors, similar to the ones used in physicochemical and ADMET predictive models or in atomic embeddings in graph-based models, as seen in various other techniques described in this paper. In RASPD+ [136] they calculated molecular weight, number of hydrogen bond donors and acceptors, logP, molar refractivity, and the Wiener topology index for the ligands and molar refractivity, logP, hydrogen bond donor and acceptor counts and binding pocket volume for selected protein residues. The residue selection was done using various distance cutoffs depending on the type of calculated descriptor. Therefore descriptors like hydrogen bond donating and accepting groups are calculated relative to the corresponding protein or ligand structures for ligands and protein residues respectively, making them more specific towards protein-ligand interactions than their more general use in other models described in Sections 4.2 & 4.4.

Another type of flat 2D embedding is the one that uses spectral graph properties of protein-ligand binding molecular graphs. PerSpectML [137], FPRC-GBT [138], AGL-Score [140], PPS-ML [141] and Nguyen et al. (2018) [139] use such properties to embed the protein-ligand interaction information. For this, they convert first a 3D protein-ligand complex representation into a graph where the nodes represent the atoms and edges of any form of inter-atom interactions. They use both inter-atomic distances and inter-atomic electrostatic energies between protein and ligand atoms and they specifically exclude protein-protein and ligand-ligand inter-atomic interactions. From this representation, various sub-graphs are defined based on different cutoff values of these inter-atomic distances or electrostatic energies. These sub-graphs can range from simple node cloud points to complex connected sub-graphs. For each sub-graph, graph properties, such as the sum of eigenvalues and their absolute deviation, spectral moments and spanning, are computed from their Laplacian matrices which give information on the graph connectivity. These properties at different cutoff levels are then converted to 2D flat feature vectors to be used in ML models.

The difference in FPRC-GBT [138] is that while in PerSpectML [137] the sub-graph properties are calculated by establishing Vietoris-Rips complexes [158], in FPRC-GBT [138] this is done by establishing Ricci curvatures. Both are types of connected graphs that exhibit specific graph properties. Further, in PPS-ML [141] they show that this can also be done through paths that are persistent across different distance thresholds and defined on the sub-graphs Laplacian matrices. In Nguyen et al. (2018) [139] they apply combined approaches using different types of complexes and compute graph properties for different graph subsets that are each focused on different atom-type interactions. Also, they use an ensemble approach where graph spectral information is fed to a random forest model together with topological information through a CNN model whose outputs were further concatenated to form the final prediction. In AGL-Score [140] the sub-graphs are established in a similar manner as in Nguyen et al. (2018) [139] whereby inter-atomic interactions are modeled at different distance thresholds for different atomic element pair subsets. However, different from Nguyen et al. (2018) [139] is that they use simple Laplacian and adjacency matrix features calculated from their eigenvectors and values for each atomic element pair sub-graphs.

Similarly to these and to models using interaction fingerprints is ^{SYBYL}GGL-Score. In ^{SYBYL}GGL-Score they use graph colouring to define subgraphs of specific protein and ligand atoms and construct 2D fingerprints based on the interatomic distances between protein and ligand atoms in these subgraphs. The subgraphs are defined based on combinations of specific protein and ligand atoms which are defined based on protein atom names and ligand SYBYL atom types respectively. The interatomic distances can be modelled using exponential or Lorentz functions which make that protein and ligand atoms that are located further away from each other will receive a zero weight to their interaction. The constructed 2D fingerprints can then be further coupled to ML models such as XGBoost. Both SYBYL ligand atom types and ECIF ligand atom types that are constructed based on interaction fingerprints between the protein and ligand atoms were tested to define the ligand atom types where the SYBYL atom types gave better performance.

A first example of how 3D representations can directly be fed into ML models to learn their embedding and a mapping to the binding affinity property is by using 3D voxel representations of the binding complex [109,143–146] similarly as to BindScope [108] with the main difference being the output of the model where in this case it returns absolute binding affinity values. This means that the models are trained in a regression setting employing loss functions such as mean absolute error. Further, Francoeur et al. (2020) [145] & AK-Score [146] both test ensemble models composed of multiple trained replicas of the same model architecture using different starting seeds for the weights and biases. They report improvement in performance compared with using a single model. Another way how 3D representations can be used directly is through graph-based models as also seen in Sections 4.2.1, 4.2 & 4.3.1. Hereby, a similar embedding principle is applied as presented in Section 4.3.1 by featurizing nodes and edges and passing these graph representations through graph-based ML models.

In AEScore [147] atomic information vectors consist of atomic environment vectors which are made up of atomic feature representations as used by the ANI model [159], which is a neural network potential model trained to predict forces and energies of small molecules. For this, the model generates atomic features based on the atomic elements and includes radial and angular information from neighboring atoms. They further use the same architecture as the original ANI model which consists of separate neural networks for each atom type with a final pooling operation to retrieve the predicted binding affinity. Interestingly in ECIFGraph::HM-Holo-Apo [149] they use two input graph representations corresponding each to unbound protein-water interaction networks and protein-ligand-crystal water-bound interaction networks. They use statistical potentials to estimate water placement using the HydraMap tool [160]. This way, additional information on desolvation and water replacement effects can be incorporated together with crystal water-bound mediated interaction information. Such desolvation effects are important since they can infer entropic contributions to the binding, which, using previously described methods, is not possible since only a single static protein-ligand pose is used whereas entropic contributions can only be inferred from dynamic features that represent protein, ligand and solvent movements during binding. Still, a drawback to this method is that both the protein and ligand are kept static. Therefore, the dynamic information of protein and ligand conformational changes upon binding gets lost. Possible ways to include them are through trace atomic information calculated from molecular dynamics simulations, such as in the new MISATO dataset [40], energy differences between unbound and bound states [132], or through augmentation of the static single binding poses using molecular dynamics simulations [161].

Performance comparison: Different standard benchmark test sets, like the Comparative Assessment of Scoring Functions (CASF) versions 2007 [162], 2013 [163,164] and 2016 [165], the Astex diverse set [166] or the Community Structure-Activity Resource (CSAR) test sets [167–170,171,172], exist for absolute binding affinity prediction, which make it possible to compare different models and methodologies between each other. In Table 4 we provide an overview of the best-performing models for the different methodologies discussed in Sections 4.3.2 & 4.3.2 on the most used benchmark test sets. Results for other benchmarks were omitted due to a high number of missing values.

First, it is clear that an objective comparison is difficult to realize from compiled literature sources as the different models have not always been tested on the same benchmark datasets. Looking at performances on the CASF2016 benchmark test set [165] the ^{SYBYL}GGL-Score method [142] could be established as the best performing model closely followed by the ECIF-LD-GBT method [120]. Interestingly hereby is that both ^{SYBYL}GGL-Score [142] and ECIF-LD-GBT [120] are not complex neural network models but simple XGBoost models linked with 2D custom-designed fingerprints either built from protein-ligand interactions in coloured subgraphs or by incorporating interaction information through interaction fingerprints and additional ligand-specific structural information. This trend has also been observed in previously described pKa and physicochemical properties prediction models (Section 4.2).

Further observing performances for the other methods we can see that many lay within a very small margin and therefore could be said to have very comparable performances. This means that different methodologies and combinations in terms of input structure embeddings and ML models provide very similar results and performances. On one hand, this can indicate that most of these methods embed similar types of information and learn similar data relationships in different ways. On the other hand, it can also indicate that the existing test sets like the CASF benchmark sets, are not difficult enough to highlight important differences in performance between the different methods. This could be due to the fact that the CASF benchmark test sets are constructed from random selections of the largest target clusters in the PDBBind training sets. The use of such random selection has already been brought up in various research [173] and the dangers of overestimating the model's performances. Therefore, methods are not tested on their generalizability capability but merely on the success of training on the PDBBind training sets. It is therefore crucial to test methods on different benchmark test sets that contain unseen information than what is present in the training sets. Therefore, testing strategies as employed in Yin et al. (2023) [157], ChemBoost [115], DeepFusionDTA [116], SimCNN-DTA [119], AttentionDTA [117] or DeepDTA [118], where testing is performed on random selections from the training set or through cross-validation, are not advisable. In general, cross-validation should be only employed during hyperparameter optimization of the ML models with final testing to be performed on held-out external test sets to avoid model construction bias towards the used test sets which can affect the model's generalizability.

Table 4
Performance comparison of some methods on the most used benchmark test sets for absolute binding affinity prediction. An asterisk indicates differences in the reported number of data points with the number of data points in the original test set.

Method name	CASF07	CASF13	CASF16
^{SYBYL}GGL-Score [142]	0.834	0.848	0.873
ECIF-LD-GBT [120]			0.866
PerSpectML [137]	0.836	0.793	0.840
PPS-ML [141]	0.836	0.793	0.840
FPRC-GBT [138]	0.831	0.805	0.834
AGL-Score [140]	0.830	0.792	0.833
ET-score [123]			0.827
Boyles et al. (2019) [134]	0.736	0.840	0.826
KDeep [109]			0.82
Wojcikowski et al. (2018) [128]		0.77	0.82
ECIFGraph::HM-Holo-Apo [149]			0.820
BAPA [122]		0.771	0.819
OnionNet [127]		0.782	0.816
AK-Score [146]			0.812
3D-RISM-AI [131]			0.80 *
AEScore [147]		0.76	0.80
Francoeur et al. (2020) [145]			0.80
Δ_{vina}XGB [132]			0.796
Fujimoto et al. (2022) [135]			0.79 *
Pafnucy [144]		0.70	0.78
GAT-Score [148]		0.78	0.776
GXLE [133]			0.762
Zhu et al. (2020) [126]			0.75
PotentialNet [130]	0.822		
SMPLIP-Score [124]		0.771	

As seen from Table 4, various benchmark test sets exist with still many others, such as the benchmark test sets, like the Schrodinger benchmark set [174], the J&J benchmark test set [175] or the Merck benchmark test set [176], used to benchmark free energy perturbation (FEP) methods, or the different binding affinity prediction challenges like D3R [177] or the Statistical Assessment of Modeling of Proteins and Ligands (SAMPL) challenges [178], which can provide out-of-distribution tests which align more closely to real case virtual screening scenarios in drug discovery. Hereby, it is also important to evaluate the performance of these models not only across different test sets but also across target-specific test sets as performances can change heavily on different targets as seen in SMPLIP-Score [124], KDeep [109], GXLE [133] and AEScore [147]. For this, the various FEP benchmark sets [174–176] have different target-specific subsets with multiple ligands docked to the same binding pockets. Such benchmark sets can also be found in the BindingDB dataset [48].

Alternatively is it also possible to cluster targets in other benchmark test sets like the CASF test sets [162–165] and evaluate on the highest populated target clusters. Potential disadvantages hereby can be that the targets in the clusters may not be completely the same. This could be avoided by setting higher clustering thresholds, risking hereby that the clusters may become sparsely populated. Another drawback to this method is that one can risk evaluating binding affinity predictions for the same or similar targets but on different binding pockets, which can also introduce bias and provide less accurate or less informed evaluations.

Further, apart from choosing multiple and well-composed benchmark test sets, one must also include various evaluation metrics apart from the classical ones like mean absolute error or Pearson's correlation. These traditionally used metrics are taken over the whole test dataset and assigned equal weights to each data point. While they are good to obtain a general notion of the model's performance for the particular test set, there are other important aspects that are not reflected by these metrics. Binding affinity prediction models are normally designed to be used in further virtual screening campaigns to find potential good binding ligands out of a large set of compounds. Metrics like early enrichment factors, which focus more on the percentage of real high binders the model ranks among its top predictions, may reflect better the model's performance in such real-case scenarios. For example, in PotentialNet [130] & ECIFGraph::HM-Holo-Apo [149] such additional metrics are used alongside the traditional ones.

Also, training sets such as the PDBBind [47] and its test sets [162–165] are constructed from high-quality crystallographic protein-ligand poses. However, during virtual screening campaigns, it is more common that compounds are docked into the target's binding pocket, as obtaining crystal poses is a time, effort and cost-expensive task and impossible to perform in a reasonable amount of time for large compound screening libraries. Therefore, binding affinity models should also be both trained and tested on re-docked or minimized binding poses as performances may drop depending on the quality of the obtained docked or minimized poses [124] and different docking algorithms can also affect binding affinity prediction models' performance [132].

Lastly, Li et al. (2022) [179] shows additional tricks on how learning binding affinity prediction models can be improved by dividing the training dataset into smaller parts and constructing independent learners on each training subset. Each of these learners optimizes independently its hyperparameters based on combined loss functions that take into account performance within each individual subtask and also across the different sub-training instances.

4.3.3. Relative binding affinity prediction

One of the difficulties in training ML models for binding affinity is the lack of high-quality structural data available. Observing widely used datasets for this task in Section 3 we can note that sizes do not go higher than around 20000 data points. When we look at models trained in other tasks such as image or text, we can see that datasets there can be in the millions to billions of data points. Thus, data scarcity is one of the major bottlenecks to achieving better performing models for tasks such as binding affinity. One of the ways to reduce this problem is by using data augmentation techniques where new data points are generated as task-meaningful modifications of the original input data. These modifications, when carefully chosen, can add additional information about the data to the model, improving its performance. One of the possible techniques to augment data in binding affinity predictions is by trying to predict the relative binding affinity between pairs of bound complexes.

In DeltaDelta [150] a 2-legged KDeep [109] model is used whereby each leg consists of the standard KDeep 3D voxel-based CNN model described in Section 4.3.2 for the parallel embedding of both protein-ligand complexes and merging of their latent embeddings to output the prediction for the relative binding affinity.

In Gusev et al. (2023) [151] they instead use 2D embedding vectors made up of path-based, Morgan [95], 3D molecular, PLEC and combinations of 3D and PLEC fingerprints. All of these describe and embed the structural and interaction information between the protein and ligand. They also employ an automated active learning cycle coupled with quantum mechanical calculations of the $\Delta\Delta G$ that serve as highly accurate estimations of the relative binding affinity which are then used to train ML models. During the automated cycle they also automatically search for the best model hyperparameters and model types from a selection of random forests, multilayer perceptrons, linear regression, k-nearest neighbors, SVM and Gaussian process models where the best performing model is then used to screen a large library of possible ligands, whereby the best scoring and more diverse ligands are fed to the quantum mechanical calculations for accurate validation and incorporation into the training set to improve the selected ML model. As the models do not use any multi-leg architectures like in DeltaDelta [150], this setup can only be used to compute binding affinity differences between the new potential ligands and an established reference ligand whose, preferably, crystallographic pose is used for its initial embedding. This means that the best selected ML model could be different depending on the target, initial reference ligand and initial training set. Therefore it can be wise to initialize the automated active learning loop with different seeds to be sure that the optimization does not get stuck in a local optimum. As it can also be only used against a specific reference ligand, this makes the ML model less general than the one used in DeltaDelta [150] which can use different reference ligands and extract additional information from pairwise differences with other good and poor binding ligands with experimentally established binding affinity values.

4.4. ADMET properties

Finally, in order to obtain a successful lead molecule, it is crucial to optimize the molecules for ADMET properties. Not doing so, one risks failure in later stages of the drug development process [180], losing valuable time and resources. When screening for ADMET properties, many different tests and biomarkers need to be considered that can provide insights into the different parts of ADMET and information on various ADMET assays are available in various public datasets as discussed in Section 3. Below we first present an overview of some important properties and biomarkers used to establish ADMET endpoints followed by a discussion of different types of ML models that can be used to predict these properties.

ADMET endpoints: Absorption constitutes the passage of the drug after intake into the systemic circulation. This usually can be expressed as the human oral bio-availability or the percentage of the drug that is found in the systemic circulation after intake [181]. The administration can also be expressed as the area under the curve (AUC) of the plasma bio-availability of the drug [182]. Hereby the Cmax [182] or the maximal achieved concentration of the drug in the systemic circulation is important as it is then when the drug can exert its effect. Low Cmax levels can indicate reduced concentrations in the target tissues and, consequently, failure of the drug to engage with the target [181]. From the other perspective, too high levels can result in toxic effects of the drug [181]. One of the factors that influences the absorption of orally administered drugs besides physicochemical factors is its transit through the membrane cells of the gastro-intestinal tract. Various assays have been designed to study the transit of drugs through this membrane such as the CACO-2 model [183], PAMPA assay [184] or the MDCK model [185]. An important factor that can influence the passage through the membrane is the action of efflux transporters such as P-glycoprotein (Pgp), which are present in many human epithelial cells [186]. These are located on the cell membrane and are responsible to pump foreign substances out of cells.

The following important property after absorption is the distribution of the drug to the target tissues. One of the indicators that can be measured for this is the apparent volume of distribution [187]. This metric measures the degree of the drug's distribution to tissues within the body out of systemic circulation. It is calculated after intravenous injection by dividing the total amount of drug administered by the blood concentration extrapolated to time 0. The distribution of the drug towards target tissues can be hampered by non-specific binding to plasma proteins such as albumin, intracellular proteins and glycoproteins [181]. Another metric important for drugs acting on the central nervous system is the permeation through the blood-brain barrier [188].

The metabolism of a drug is the transformation of the drug by enzymes in the body to its metabolites. This process is important for several reasons [181]. First, it is an important step in the elimination of some drugs from the body by transforming them into metabolites that can be more easily excreted through the bile or urine. Second, some drugs that are precursors depend on metabolization to become active. Third, metabolism also plays an important role in toxicity as some reactive metabolites can produce adverse toxic effects. The different enzymes involved in metabolism, mostly from the CYP enzyme family [189], are used as bio-markers to test the drug's metabolism. Liver microsomes [190] are another important biomarker for metabolism. These are vesicles found in the endoplasmic reticulum of the hepatocyte and contain various expressed phase I and II metabolic enzymes such as CYP-enzymes, flavine monooxygenases, esterases, amidases, epoxide hydrolases and UDP glucuronyltransferases. Another important bio-marker is the induction of the Pregnane X receptor (PXR) which in turn induces the expression of genes coding for CYP enzymes, conjugation enzymes such as carboxylesterases and transporters like MDR1 [191–193].

Finally, excretion is the elimination of the drug from the body [181]. This happens through one of the body fluids, gases or hair, and either directly the unmodified drug is eliminated or its metabolites. Excretion is often measured as clearance [181] which is the rate at which the drug is removed from the plasma divided by its plasma concentration. Total clearance [194] is the combination of all the clearance pathways by which the drug can be eliminated. Linked to this is the half-life [181] of the drug or the time necessary to reduce the drug's plasma concentration by half.

Toxicity reflects a drug's action that does not form part of its intended mode of action. It can be classified under several types such as [195]: (1) on-target toxicity involving its main target of action, (2) hypersensitivity and immune responses that occur due to interactions of the drug with targets that induce these immune reactions, (3) off-target toxicity, where the drug binds to other targets than its main intended targets, (4) bio-activation whereby the drug's metabolites interact with proteins in the body and cause toxicity through immune reactions or off-target toxicity, (5) idiosyncratic reactions that are rare, not well understood and more specific to individuals. Some targets are generally used in toxicity screening assays due to their importance such as the potassium channels encoded and regulated by the human ether-á-go-go-related gene (hERG) [196] for cardiotoxicity screening, targets involved in hepatotoxicity [197] or tests like the Ames mutagenicity test [198] that probes if the drug can cause alterations to the DNA, important for mutagenicity screening, or the micronucleus test [199] for genotoxicity.

ADMET predictive ML models: Different models exist for ADMET properties prediction that can be seen in Table 5. Similar to models used for physicochemical properties prediction (Section 4.2), these models also use whole molecule embeddings. For this, they can use both 2D feature vector embeddings constructed from molecular properties like in chemical embeddings and structural fingerprints or they can use molecular network graph embeddings.

For the construction of 2D feature vectors, several chemical descriptors can be used such as molecular weight, hydrogen bond donating and accepting groups, polar surface area or drug-likeness measures [215], which all can be easily calculated through packages such as RDKit [97]. Some [215,214,206,91,213] are further extended with structural descriptors like Morgan fingerprints [95], MACCS keys [155], atom-pair counts or descriptors [93] or ECFPs [154]. These structure-based descriptors can also be used alone [209]. Or the chemical descriptors can also be extended with computed ADMET or physicochemical properties such as logP/logD, solubility, clearance, metabolic information or cellular permeability [203,207,204–206].

Often these sets of descriptors are further reduced by removing highly correlated features, features with many missing values or that show very low variance between compounds. Specific algorithms, such as the Boruta algorithm [216] can be additionally used to estimate feature importance and select only the most relevant features. This reduction helps to avoid unnecessary features to keep computational load efficient and can improve model performance by removing inter-correlated features which can interfere with training [214].

In Orosz et al. (2022) [214] and Doweyko (2004) [217] they observed an interesting drop in performance when combining multiple descriptor types such as both 2D and 3D chemical and structural descriptors. In sharp contrast, Yin et al. (2023) [157] found an improvement in performance for absolute binding affinity prediction models when fingerprints encoding different information types, such as structural molecular information and interaction information between targets and their ligands, were combined. Likewise a combination of different structural descriptors in the BTAMDL model [209] gave a better performance than using them individually. Possibly as each structural descriptors takes different structural molecular information

Table 5
Overview of ADMET models. Methods indicated with an asterisk use multi-task learning or a combination of single and multi-task learning.

Method name	Embedding	Tested models	Properties
Regression Models			
Siramshetty et al. (2021) [200]	CD	RF, **GCN**	RLM stability assay, PAMPA, KAS
MMPA-by-QSAR [89]	CD	**RF**	lipophilicity, HLM
Zhu et al. (2018) [201]	2D/3D CD	MLR, SVM, MARS, **RF**	Blood-brain partitioning
Zhou et al. (2019) [202]	8192-bit ECFP	DNN, SVM	HTSA solubility, CYP3a4/CYP2c9/CYP2d6, microsomal metabolic stability, Pgp, MDCK cell permeability, unbound fraction in microsomes/brain/plasma
Wenzel et al. (2019) [91] *	CD + atom-pair and pharmacophoric donor-acceptor pair desc.	**DNN**	$Cl_{m}et$, Caco-2, metabolic liability, logD
Kosugi et al. (2021) [203]	CD + exp. ADMET props.	RF, GP	HOB
Obrezanova et al. (2022) [204]*	CD + exp. ADMET props.	**GCN**, GP, XGBoost, SVM, **DNN**	F, $Cl_{t}ot$, $Vd_{s}s$, AUC, $C_{m}ax$, HL, CT-curves
Kosugi et al. (2020) [205]	CD + ADMET props.	PLS, **RBF**, **RF**, GP	$Cl_{tot,rat}$
Yuan et al. (2020) [206]	2D/3D CD & SD + ADMET props.	kNN, SVM, RF, boost tree, GBR, **ensemble**	PPB
Miljkovic et al. (2021) [207]	CD + pred. ADMET prop. + dose	RF, XGBoost	AUC, HOB, $C_{max, plasma}$, $Cl_{r}en$, $Cl_{t}ot$, HL, $t_{c}max$, $Vd_{ss/IV}$
Chemi-Net [87]*	Atom/bond desc.	**GCN**	AqSol, CYP3a4, HLM, HOB, PXR
Broccatelli et al. (2022) [88] *	Atom/bond desc.	GCN, **GAT**, MPNN, AttentiveFP	logD, $Cl_{HLM/hepatocytes}$, kinetic solubility in phosphate buffer
Lim et al. (2022) [208]	atom/ bond descr., QM9 pred. props., CD, ANI-2x energies	**GCN**	rat hepatocyte, rat and human microsome, rat Cl_{tot}, rat and human Pgp
BTAMDL [209]	SD	**GBDT with DNN**	$LD50_{rat}$, IGC50, LC50, LC50DM
Classification Models			
Li et al. (2023) [210]	CD	**LXGBoost**, PLS DA, AdaBoost	caco-2, CYP3a4, hERG, HOB, Micronucleus test
ABERT [211]	CD	**ABERT**, DT, RF, ERT, feed forward NN, RESNET	caco-2, HOB, CYP3a4
Falcon-Cano et al. (2020b) [212]	CD	XGBoost, SVM, DT, MLP, naive Bayes, **Ensemble**	HOB
Zhou et al. (2023) [202]	8192-bit ECFP	DNN, SVM	HTSA solubility, CYP3a4/CYP2c9/CYP2d6, microsomal metabolic stability, Pgp, MDCK cell permeability, unbound fraction in microsomes/brain/plasma

(continued on next page)

Table 5 (*continued*)

Method name	Embedding	Tested models	Properties
Chen et al. (2023) [213]	CD + SD	kNN, **SVM**, **RF**, feed-forward NN, **GCN**	hERG
Orosz et al. (2022) [214]	2D/3D CD & SD	**XGBoost**, FFNN	Ames, Pgp inhibition, hERG, hepatotoxicity, BBB permeability, CYP2c9
AECF [215]	CD, SD, drug-likeness desc.	DA, SVM, FFNN, RF, max likelihood, nearest centroid, kNN, **ensemble**	Caco-2, HIA, HOB, Pgp binding type
Yuan et al. (2020) [206]	2D/3D CD & SD + ADMET props.	kNN, SVM, RF, boost tree, GBR, **ensemble**	PPB
GGL-Tox [73]	Graph colouring using protein atom names and SYBYL ligand atom types	**GBDT**, SVM, RF	endpoints present in Tox21 dataset

Abbreviations embeddings: CD = chemical descriptors, SD = structural descriptors, props. = properties, desc. = descriptors. Abbreviations models: kNN = k nearest neighbours, SVM = support vector machine, GCN = graph convolutional network, XGBoost = extreme gradient boosting, AdaBoost = adaptive boosting, ABERT = adaptive boosting extremely random tree, RESNET = residual network, MLP = multi layer perceptron, GAT = graph attention network, MPNN = message passing neural network, FP = fingerprint, MLR = multivariate linear regression, MARS = multivariate adaptive regression spline, RF = random forest, LXGBoost = light XGBoost, PLS = partial least squares, DA = discriminant analysis, DT = decision trees, ERT = extreme random trees, NN = neural network, DNN = deep neural network, GP = gaussian processes, RBF = radial basis functions, DA = flexible discriminant analysis, GBR = gradient boosting regression, FFNN = feed forward neural network, GBDT = gradient boosted decision trees. Abbreviations properties: HOB = human oral bioavailability, KAS = kinetic aqueous solubility, HLM = human liver microsomes, AUC = area under time plasma concentration curve, HL = elimination half life, HIA = human intestinal absorption, AqSol = aqueous solubility, PPB = plasma protein binding, Cl_tot = total clearance, $C_{max,plasma}$ = peak plasma concentration, t_cmax = time to peak plasma concentration, $Vd_{ss/IV}$ = volume of distribution at steady state or after IV administration, CT-curves = concentration-time curves, $Cl_{HLM/hepatocytes}$ = clearance in human liver microsomes or hepatocytes, Cl_met = metabolic clearance, LD50 = lethal dose 50, LC50 = lethal concentration 50, IGC50 = 50% inhibitory growth concentration

into account. This points out the importance to ensure that selected descriptors are relevant and highly informative for the prediction task and property at hand. This could potentially be evaluated through entropy-based techniques such as information gain used in decision trees. Also, an improvement in predictions was observed in Kosugi et al. (2021) [203] by incorporating experimental results of relevant molecular properties. While one can also use computed property values, experimental results generally have a lower error and would produce more accurate embedding descriptors.

Another group of models uses molecular graph information with either simple ML models such as gradient-boosted decision trees [73] or with more complex graph-based neural networks [208,87,88]. The former employ for this a graph colouring technique similar to the GGL-Score model [142] described in Section 4.3.2. The latter employ atom and bond-specific chemical descriptors on constructed molecular network graphs, such as those described in Section 4.2, since they model each atom and bond explicitly.

Performance of ADMET predictive ML models: Performance-wise, a wide array of ML models has been used and tested with either type of molecular embedding, ranging from simple models such as decision trees, random forests, XGBoost, SVM, kNN, MLR, MARS, partial least squares, radial basis functions, Gaussian processes to more complex neural network models like feed-forward neural networks and graph-based neural networks. In general, no clear advantage of one model over the other can be observed with performance often being very comparable between the different models [207,203,202] and performance also being very dataset dependent. In addition, several neural network models [213,211,202] did not seem to improve simpler models and showed overfitting for some smaller datasets [213] in which setting, simpler models could be more beneficial. The GGL-Tox model further also showed an interesting way to combine molecular graph-based embeddings with simple gradient-boosted decision trees ML model, reporting good performance across different Tox21 sub-datasets. This could be an interesting middle-ground solution to combine complex types of molecular featurization with more simple ML models which also showed good performance in other molecular properties prediction tasks such as binding affinity (see Section 4.3.2).

Some [212,215,206] experimented with ensemble models by combining predictions from single model architectures. They reported improved performance over their single-model counterparts. AECF [215] used hereby a genetic algorithm to select the best combinations of datapoints sampling, individual ensemble models and ensemble aggregation rules.

Many [210,211,214,208,89,202,207,204,215,87,88,91,73,209] constructed models for several ADMET properties. However, with the large amount of assay data available in datasets like Tox21 [68,69] or CHEMBL [43], to the best of our knowledge, no model or application currently exists that would combine models for all possible and available ADMET properties. One possible reason could be the lack of a sufficient number of data points for some of the properties, as dataset sizes could be below 200 or even 100 data points. This is insufficient to train accurate and general models. Also, much of the ADMET data is often private and part of drug discovery projects in pharmaceutical companies [208,89,202–205,87,88,91]. Their models are constructed based on project needs and possibly not all ADMET endpoints are of major importance.

While many constructed separate models for each property, some also tested a multi-task learning approach where one single model was trained on several ADMET properties, either in parallel or sequentially. The sequential approach is usually favored when compounds in the different subsets have a low degree of overlap [91]. As this capability is native to neural network models, which are highly flexible in their architecture design, such approaches were not seen using simpler models alone such as random forests or SVMs. An interesting solution here was presented in the BTAMDL model [209] where a multi-task trained deep neural network was combined with a gradient-boosted decision trees model by using the learned latent vector from the last layer of the DNN as the input vector to the GBDT model. Using this approach they noted an improved performance over using the DNN or GBDT models alone for 3 out of 4 learning tasks. As mentioned previously in Section 4.2.3, such a multi-task learning approach did not always perform better compared to its single-task counterpart as seen in Chemi-Net [87] & Wenzel et al. (2019) [91]. In Wenzel et al. (2019) [91] they noticed that combining highly orthogonal properties does not always result in improved performance when training in a multi-task setting. This was also seen in Broccatelli et al. (2022) [88] where only complementary ADMET properties were trained together. Still, when combined well this could potentially improve training on small datasets.

Lastly, there is also an even distribution between classification and regression models. Some models like Yuan et al. (2020) [206] or the one from Falcòn-Cano et al. (2020a) [86] combined classification and regression models to construct more accurate regressions that would span a smaller range of values. Hereby, classification models serve to separate compounds into one of the value ranges before generating a more accurate prediction with the regression models. In Zhou et al. (2019) [202] such data splitting was also performed without the prior use of classification models. They also compared the regression models with their classification counterparts and observed a higher robustness of the latter under scaling of the prediction values.

4.5. Understanding predictions

An important aspect when developing ML models is their interpretability, especially for models involved in critical decision making like those employed in drug discovery. Interpretability methods can help to understand why the model is making certain predictions for the corresponding input data, giving validity to the predictions. They can also help to uncover hidden bias or errors in the model and hence can also assist in model development and optimisation. An important aspect of these techniques is that these methods should be consistent, accurate, faithful and stable [218] in order to provide correct interpretations. The different modeltypes described in this work all have different possible interpretability techniques and some are easier to interpret than others.

Linear models and support vector machines (SVMs) are simple models and are therefore easier to interpret than other more complex models. For linear models, standardized feature weights used to construct the linear function can be used to indicate the importance of each feature or one can observe changes in either the outcome value or the variables when changing the value of the other respectively. SVMs are slightly more complex because they transform the input feature space into higher dimensions. Platt scaling [219] can be used here to perturb each feature of the input data point and output it to probabilities of feature importance.

Decision trees, random forests and XGBoost models can usually be interpreted through techniques like Saabas. Saabas technique [220] tries to interpret the model as a linear combination of features and the decision rules that were applied to get to the final value. This can produce feature importance plots that visualize which features have a higher weight to get to the predicted values. While the technique is easy to use, it can suffer from low consistency [221]. Another technique that can be used are Shapley values [222]. This method assigns importance values to each feature and can be more reliable in terms of consistency and accuracy of the explanations. Such interpretability can prove to be very useful to validate that the model captures meaningful correlations. For example in Yuan et al. (2020) [206] it was seen that lipophilicity-based descriptors in the feature vector had the highest importance for predicting plasma protein binding while in Zhu et al. (2018) [201] the most important features to predict blood-brain partitioning were the topological polar surface area, log octanol-water partition coefficient, van der Waals polar surface area, number of hydrogen bond donors and solvation energies. All features that clinically are also highly correlated with their prediction values.

Neural networks are larger and more complex models and are therefore harder to interpret because of the high non-linearity and the number of parameters and are often termed “black box" models. Nonetheless, various techniques exist to analyze different parts and aspects of these models [223]. The first way is to analyze directly the weights of the network. This often is difficult for large models due to the high dimension and number of hidden layers. It also does not take interactions between the hidden neurons of the network into account. Another way is by looking at the activations of the neurons and their outputs. This takes into account layer interactions but can again be difficult to interpret for large networks due to their size and high dimensionality. One way to circumvent this is by applying dimensionality reduction techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) [14] or Uniform Manifold Approximation and Projection (UMAP) [224] to the outputs of the layer, such as in the final embedding layers, and project this lower dimensional output in two or three dimensions. These can further be combined across different data points to get a global understanding of how the model behaves on various input data [225]. Sometimes it can however be also interesting to visually see what parts of the input data contribute higher to the obtained prediction, especially for visual inputs like images or graphs. To do this, saliency maps [226] can be constructed that consist of the original input data together with mapped importance values for each part of this input space. These importance values are obtained by backpropagating the gradients of the predicted output through the network on to the input features for different levels of modified inputs. Graph neural networks can be tricky to interpret effectively as they have connectivities both in their graph input data between the nodes as well as throughout the graph model between the neurons. Layerwise relevance propagation (LRP) together with combined graph/model walks [227] have been shown to be able to capture and visualize this complex connectivity. Here, neuron relevances are calculated and backpropagated according to set rules using combined graph/model walks. This produces graph saliency maps highlighting the important nodes and edges that contribute the most to the prediction. This method however, can become computationally very expensive for large input graphs and models. A simpler approach for chemical graph data is the use of counterfactuals [228] where modifications to the input chemical structures are applied for which predictions are generated. These are then compared to the original input to provide an understanding of the importance of different chemical groups in the input structure for the generated predictions. On top of that it can, at the same time, produce important information for further lead optimisation of the screened compounds.

5. Conclusion

In this work we have shown the wide array of possible ML models and methods for small molecular properties predictions that can be used in drug discovery virtual screening campaigns. While the different methods use slightly different amounts and types of input information or transform them through different techniques, the reported performances lay often very close, with very limited differences in performance, which for practical applications is not too relevant. We also see that, while more complex, more flexible and computationally more expensive, neural network based models are not always able to outperform their simpler counterparts in the current context of generally low data regimes. Although, their higher flexibility can be exploited in interesting ways such as through multi-task learning in, for example, ADMET prediction models (Section 4.4). But, this does not always provide the better results [88]. Therefore, additional effort should be taken to validate its training strategy in order to establish best practices, and care needs to be always taken when selecting the different subtasks, as these need to be complementary to ensure successful training and gains in performance.

While this lack of difference between the models could be attributed to the fact that all of them learn very similar relationships in the data through either explicit or implicit ways using other surrogate descriptors, it could also highlight a lack of strong and diverse benchmark test sets. While there exist various benchmark sets for the binding affinity prediction task, it was shown that many are not widely used, that there is a lack of consensus on which benchmarks to include during model testing and that some, like the CASF [165,162–164] test sets, resemble too closely the training data.

For the other property prediction tasks good standard benchmarks seem to be absent or not widely used, as many report test sets taken from training data or by cross-validation performances. This, as explained earlier, can result in tests that are too representative of the training data and would therefore not provide results on the generalizability of the ML models. These issues raise a need for the establishment of better, stronger, standardized and widely accepted benchmarks. These benchmarks should provide a correct balance between in and out of domain molecules [212] in order to test the model’s generalizability without also underestimating its performance through too difficult test sets that are not reflective anymore of the real case scenarios in which these models will be used. Test sets should therefore reflect as closely as possible data found in the real case scenario applications. This can be, for example, usage of time-based splits [91,88,204,205] instead of random compound selections as it has been shown [173] that both random selection and held-out compound clusters both over-or underestimate the model's performance respectively. Of course, time information is not always accessible. That is why techniques such as simulated medicinal chemistry project data (SIMPD) [173] can help to establish datasplits that are highly similar to time-based splits in drug discovery projects.

Other ways to construct test sets could also be based on the inclusion of compound-dose related combinations like in Miljkovic et al. (2021) [207] for properties that are also dose dependant, in order to test the model's sensitivity to interpret and use such data correctly. Further, it can also help to base the compound selection for testing on scaffold clusters [204,205] in order to ensure a wide, heterogenous selection of structurally different compounds for models that need to perform well on a wide variety of molecules. Initiatives such as the Huggingface platform for language models and research or the Therapeutics Data Commons [45,46] in biomedical research could also help to guide the community towards this needed standardization by collecting and providing validated training and test datasets and establishing leaderboards in order to more objectively compare newly developed models and highlight differences in performance in order to drive research further faster in the right direction.

Abundant high quality training data is also needed to be able to train high quality models. While various datasets exist (Section 3), they often lack a sufficiently large amount of datapoints. This is especially true for certain ADMET endpoint datasets which can be just in the ranges of a couple of hundred datapoints. Certain data augmentation techniques exist however to overcome the issue such as pretraining of neural networks on general molecular structural data with further finetuning on the specific property prediction datasets, multitask learning such as in several ADMET property prediction neural network models or use of molecular dynamics simulations to augment binding affinity datasets that are comprised of 3D binding complex structures, with additional bound conformations. However, besides size, quality is another important factor as large poor quality datasets still can generate models that will underperform [91]. This can, for example, be due to class imbalances in classification models [206,207]. Another problem of many biomedical datasets is the large number of missing data. This can be solved for example by labeling the data with a pretrained ML model [204] after which these newly labelled datapoints can be incorporated into the training data. One still needs to take care hereby that the generated predictions are sufficiently trustworthy. Therefore one can use multi-fidelity models by generating predictions with multiple trained replicas of the same model and use the deviation on the different generated predictions as a metric of precision. Or one can also analyze how well the predicted data points are embedded in the training data to know if the new data is in-or out of domain.

Regarding the future of ML models for the prediction of small molecular properties in drug discovery, we can highlight several interesting research directions to improve this field further. Firstly, as Bayesian models provide, apart from the predictions, also uncertainty estimates of these, they are an interesting group of ML models to investigate further as uncertainty estimation is particularly important for models used in decision support and more in critical areas such as drug discovery. Several works [203–205,151] have reported usage or tests with Bayesian models in different molecular properties prediction tasks and as Bayesian models exist both for simple and more complex neural network architectures, it should be fairly easy to combine them with different molecular featurization techniques discussed in present work. Likewise, further investigation can also be performed into other techniques for uncertainty estimation that can be used with the other ML methods. Secondly, data scarcity is an important limitation in various prediction tasks especially regarding certain ADMET endpoints. Therefore, efforts to curate larger and qualitatively better datasets will pay off in improved model performances. This also means that collaborative partnerships and data sharing could become key to collecting larger amounts of private data for training. Also, further research into multi-task learning could provide further important insights into the design of models that can incorporate multiple, diverse, small datasets as a way to augment the amount of data that can be used. Thirdly, as molecular protonation and deprotonation are characterized by both micro and macro-pKa values which are reaction rates of hydrogen exchanges at either individual protonation sites in a small molecule or the protonation or deprotonation of the complete molecule, it is important to take this complete and complex information into account. Many seen models in pKa prediction either take only micro or macro pKa information into account. However, expansion of the data with protonation states ensembles and different tautomeric forms as in the latest Epik model [85], can improve the performance of models in pKa and protonation state prediction drastically. Lastly, several properties such as those describing quantum mechanical information of a small molecule were able to improve the performance of ML models in various prediction tasks. These, however, are expensive to obtain and would thus form a bottleneck in ML-assisted virtual screening pipelines. One way to circumvent this could be by replacing them with their ML predicted counterparts. Use of 3D molecular information can also be important for several properties like in pKa or binding affinity prediction. However, using complex graph-based neural network models might increase the risk of overfitting when little data is available for training. Techniques such as in GGL-Score [142] or ECIF-LD-GBT [120] where 3D information is embedded into 2D vectors that can be combined with simpler models like tree-based models, could provide a good balance between input data and model complexity.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT (OpenAI) in order to assist in the drafting of the abstract and the introductory paragraph of Section 3 on datasets. After using this tool/ service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Funding

The project PID2020-116564GB-I00 has been funded by MCIN/ AEI /10.13039/501100011033. This project has received funding from the Torres-Quevedo Programme from the Spanish National Agency for Research (PTQ2021-011669 / AEI / 10.13039/501100011033). With the support of the Industrial Doctorates Plan of the Secretariat of Universities and Research of the Department of Economy and Knowledge of the Generalitat of Catalonia.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors of this paper are either employed by or have financial interest in the company Acellera Labs. The first author is receiving an industrial doctorate grant according to the Industrial Doctorates Plan from Secretariat of Universities and Research of the Department of Economy and Knowledge of the Generalitat of Catalonia.

References

[1] D. Austin and T. Hayford, Research and development in the pharmaceutical industry, Congressional Budget Office, Tech. Rep., 2021.

[2] European Federation of Pharmaceutical Industries and Associations, The pharmaceutical industry in figures: Key data, European Federation of Pharmaceutical Industries and Associations, Tech. Rep., 2022.

[3] H. Dowden, J. Munro, Trends in clinical success rates and therapeutic focus, Nat. Rev. Drug Disc. 18 (7) (2019) 495–496, https://doi.org/10.1038/d41573-019-00074-z.

[4] I. Kola, J. Landis, Can the pharmaceutical industry reduce attrition rates? Nat. Rev. Drug Disc. 3 (8) (2004) 711–716, https://doi.org/10.1038/nrd1470.

[5] D. Bassani, S. Moro, Past, present, and future perspectives on computer-aided drug design methodologies, Mol 28 (9) (2023). ⟨https://www.mdpi.com/1420-3049/28/9/3906⟩.

[6] I.D. Kuntz, J.M. Blaney, S.J. Oatley, R. Langridge, T.E. Ferrin, A geometric approach to macromolecule-ligand interactions, J. Mol. Biol. 161 (2) (1982) 269–288, https://doi.org/10.1016/0022-2836(82)90153-x.

[7] N.S. Pagadala, K. Syed, J. Tuszynski, Software for molecular docking: a review, Biophys. Rev. 9 (2) (2017) 91–102, https://doi.org/10.1007/s12551-016-0247-1.

[8] T. Pantsar, A. Poso, Binding affinity via docking: fact and fiction, Mol 23 (8) (2018). ⟨https://www.mdpi.com/1420-3049/23/8/1899⟩.

[9] M. De Vivo, M. Masetti, G. Bottegoni, A. Cavalli, Role of molecular dynamics and related methods in drug discovery, J. Med. Chem. 59 (9) (2016) 4035–4061, https://doi.org/10.1021/acs.jmedchem.5b01684.

[10] I.A. Guedes, F.S.S. Pereira, L.E. Dardenne, Empirical scoring functions for structure-based virtual screening: applications, critical aspects, and challenges, Front. Pharmacol. 9 (2018), https://doi.org/10.3389/fphar.2018.01089.

[11] S. Dara, S. Dhamercherla, S.S. Jadav, C.M. Babu, M.J. Ahsan, Machine learning in drug discovery: a review, Artif. Intell. Rev. 55 (3) (2021) 1947–1999, https://doi.org/10.1007/s10462-021-10058-4.

[12] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.

[13] T. Bellotti, I. Nouretdinov, M. Yang, A. Gammerman, Chapter 6 - feature selection, in: V.N. Balasubramanian, S.-S. Ho, V. Vovk (Eds.), Conformal Prediction for Reliable Machine Learning, Morgan Kaufmann, Boston, 2014, pp. 115–130. ⟨https://www.sciencedirect.com/science/article/pii/B9780123985378000067⟩.

[14] L. van der Maaten, G. Hinton, Viualizing data using t-sne, J. Mach. Learn. Res. 9 (2008) 2579–2605.

[15] T. Hastie, R. Tibshirani, A. Buja, Flexible discriminant analysis by optimal scoring, J. Am. Stat. Assoc. 89 (2000).

[16] J.H. Friedman, Multivariate adaptive regression splines, Ann. Stat. 19 (1) (1991) 1–67, https://doi.org/10.1214/aos/1176347963.

[17] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B-Methodol. 58 (1996) 267–288.

[18] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 42 (1) (2000) 80–86, https://doi.org/10.1080/00401706.2000.10485983.

[19] A. Höskuldsson, Pls regression methods, J. Chemom. 2 (3) (1988) 211–228.

[20] C. Cortes, V.N. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[21] V. Vapnik, S.E. Golowich, A. Smola, Support vector method for function approximation, regression estimation and signal processing. Proc. of the 9th Int. Conf. on Neural Inf. Process. Syst., ser. NIPS'96, MIT Press, 1996, pp. 281–287.

[22] V. Vovk, Kernel Ridge Regression, 10 2013, 105–116.

[23] D. Packwood, L.T.H. Nguyen, P. Cesana, G. Zhang, A. Staykov, Y. Fukumoto, D. H. Nguyen, Machine learning in materials chemistry: An invitation, Mach. Learn. Appl. 8 (2022), 100265. ⟨https://www.sciencedirect.com/science/article/pii/S2666827022000093⟩.

[24] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, ser. Camb. Monogr. on Appl. and Comp. Math., Cambridge University Press, 2003.

[25] O. Obrezanova, G. Csányi, J.M.R. Gola, M.D. Segall, Gaussian processes: A Method for automatic qsar modeling of adme properties, J. Chem. Inf. Model. 47 (5) (2007) 1847–1857, https://doi.org/10.1021/ci7000633.

[26] L.H. Mervin, S. Johansson, E. Semenova, K.A. Giblin, O. Engkvist, Uncertainty quantification in drug design, Drug Discov. Today 26 (2) (2021) 474–489. ⟨https://www.sciencedirect.com/science/article/pii/S1359644620305110⟩.

[27] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and regression trees, English.1984.

[28] G.V. Kass, An exploratory technique for investigating large quantities of categorical data, J. R. Stat. Soc.: Ser. C. (Appl. Stat.) 29 (2) (1980) 119–127.

[29] E.B. Hunt, J. Marin, and P.J. Stone, Experiments in induction.1966.

[30] J.R. Quinlan, Learning efficient classification procedures and their application to chess end games. Machine learning, Elsevier, 1983, pp. 463–482.

[31] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.

[32] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[33] J. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (2000).

[34] J. Friedman, Stochastic gradient boosting, Comp. Stat. Data Anal. 38 (2002) 367–378.

[35] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386–408.

[36] T.N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, 2017.

[37] P. Veličković, et al., Graph attention networks, 6th Int. Conf. on Learning Represent., 2017.

[38] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, and G.E. Dahl, Neural message passing for quantum chemistry, In: Pser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., 70. PMLR, 2017, 1263–1272.⟨https://proceedings.mlr.press/v70/gilmer17a.html⟩.

[39] D.B. Korlepara, et al., Plas-5k: dataset of protein-ligand affinities from molecular dynamics for machine learning applications, Sci. Data 9 (1) (2022).

[40] T. Siebenmorgen, et al., Misato - machine learning dataset of protein-ligand complexes for structure-based drug discovery, bioRxiv, 2023.⟨https://www.biorxiv.org/content/early/2023/05/28/2023.05.24.542082⟩.

[41] K.M. Gayvert, N.S. Madhukar, O. Elemento, A data-driven approach to predicting successes and failures of clinical trials, Cell Chem. Biol. 23 (10) (2016) 1294–1301.

[42] D.S. Wishart, et al., DrugBank 5.0: a major update to the DrugBank database for 2018, Nucleic Acids Res. 46 (D1) (2017) D1074–D1082, https://doi.org/10.1093/nar/gkx1037.

[43] D. Mendez, et al., ChEMBL: towards direct deposition of bioassay data, Nucleic Acids Res. 47 (D1) (2018) D930–D940, https://doi.org/10.1093/nar/gky1075.

[44] S. Kim, et al., PubChem 2023 update, Nucleic Acids Res., 51(D1), D1373-D1380, 2022.10.1093/nar/gkac956.

[45] K. Huang, et al., Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development, 2021.

[46] K. Huang, et al., Artificial intelligence foundation for therapeutic science, Nat. Chem. Biol. 18 (10) (2022) 1033–1036.

[47] R. Wang, et al., The pdbbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures, J. Med. Chem. 47 (12) (2004) 2977–2980, https://doi.org/10.1021/jm030580l.

[48] M.K. Gilson, et al., BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology, Nucleic Acids Res. 44 (D1) (2015) D1045–D1053, https://doi.org/10.1093/nar/gkv1072.

[49] R.D. Smith, et al., Updates to binding moad (mother of all databases): polypharmacology tools and their utility in drug repurposing, J. Mol. Biol. 431 (13) (2019) 2423–2433 (computation Resources for Molecular Biology), ⟨https://www.sciencedirect.com/science/article/pii/S0022283619302967⟩.

[50] J. Tang, et al., Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis, J. Chem. Inf. Model. 54 (3) (2014) 735–743, https://doi.org/10.1021/ci400709d.

[51] C. Yung-Chi, W.H. Prusoff, Relationship between the inhibition constant (ki) and the concentration of inhibitor which causes 50 per cent inhibition (i50) of an enzymatic reaction, Biochem. Pharmacol. 22 (23) (1973) 3099–3108. ⟨https://www.sciencedirect.com/science/article/pii/0006295273901962⟩.

[52] A.P. Graves, R. Brenk, B.K. Shoichet, Decoys for docking, J. Med. Chem. 48 (11) (2005) 3714–3728, https://doi.org/10.1021/jm0491187.

[53] M.M. Mysinger, M. Carchia, J.J. Irwin, B.K. Shoichet, Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking, J. Med. Chem. 55 (14) (2012) 6582–6594, https://doi.org/10.1021/jm300687e.

[54] S.G. Rohrer, K. Baumann, Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data, J. Chem. Inf. Model. 49 (2) (2009) 169–184, https://doi.org/10.1021/ci8002649.

[55] V.-K. Tran-Nguyen, C. Jacquemard, D. Rognan, Lit-pcba: an unbiased data set for machine learning and virtual screening, J. Chem. Inf. Model. 60 (9) (2020) 4263–4273, https://doi.org/10.1021/acs.jcim.0c00155.

[56] N. Huang, B.K. Shoichet, J.J. Irwin, Benchmarking sets for molecular docking, J. Med. Chem. 49 (23) (2006) 6789–6801, https://doi.org/10.1021/jm0608356.

[57] S.M. Vogel, M.R. Bauer, F.M. Boeckler, Dekois: demanding evaluation kits for objective in silico screening - a versatile tool for benchmarking docking programs and scoring functions, J. Chem. Inf. Model. 51 (10) (2011) 2650–2665, https://doi.org/10.1021/ci2001549.

[58] A.C. Good, T.I. Oprea, Optimization of camd techniques 3. virtual screening enrichment studies: a help or hindrance in tool selection? J. Comput. -Aided Mol. Des. 22 (3–4) (2008) 169–178.

[59] P.C. Hawkins, G.L. Warren, A.G. Skillman, A. Nicholls, How to do an evaluation: pitfalls and traps, J. Comput. -Aided Mol. Des. 22 (3–4) (2008) 179–190.

[60] L. Chaput, J. Martinez-Sanz, N. Saettel, L. Mouawad, Benchmark of four popular virtual screening programs: construction of the active/decoy dataset remains a major determinant of measured performance, J. Chemin.-. 8 (1) (2016).

[61] I. Wallach, A. Heifets, Most ligand-based classification benchmarks reward memorization rather than generalization, J. Chem. Inf. Model. 58 (5) (2018) 916–932, https://doi.org/10.1021/acs.jcim.7b00403.

[62] L. Chen, A. Cruz, S. Ramsey, C.J. Dickson, J.S. Duca, V. Hornak, D.R. Koes, T. Kurtzman, Hidden bias in the dud-e dataset leads to misleading performance of deep learning in structure-based virtual screening, PLoS One 14 (8) (2019) 1–22, https://doi.org/10.1371/journal.pone.0220113.

[63] J. Sieg, F. Flachsenberg, M. Rarey, In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening, J. Chem. Inf. Model. 59 (3) (2019) 947–961, https://doi.org/10.1021/acs.jcim.8b00712.

[64] U.S. Environmental Protection Agency:: U.S. EPA.Physprop database. estimation programs interface suite for microsoft windows, v 4.11: Perfluorooctanesulfonic acid (pfos) (casrn 1763–23-1), U.S. Environmental Protection Agency, Tech. Rep., 2012.

[65] K. Wu, Z. Zhao, R. Wang, G.-W. Wei, Topp-s: Persistent homology-based multi-task deep neural networks for simultaneous predictions of partition coefficient and aqueous solubility, J. Comput. Chem. 39 (20) (2018) 1444–1454, https://doi.org/10.1002/jcc.25213.

[66] D. Chen, K. Gao, D.D. Nguyen, X. Chen, Y. Jiang, G.-W. Wei, F. Pan, Algebraic graph-assisted bidirectional transformers for molecular property prediction, Nat. Commun. 12 (1) (2021), https://doi.org/10.1038/s41467-021-23720-w.

[67] D. Chen, J. Zheng, G.-W. Wei, F. Pan, Extracting predictive representations from hundreds of millions of molecules, J. Phys. Chem. Lett. 12 (44) (2021) 10793–10801, https://doi.org/10.1021/acs.jpclett.1c03058.

[68] A. Mayr, G. Klambauer, T. Unterthiner, S. Hochreiter, Deeptox: toxicity prediction using deep learning, Front. Environ. Sci. 3 (2016), https://doi.org/10.3389/fenvs.2015.00080.

[69] R. Huang, M. Xia, D.-T. Nguyen, T. Zhao, S. Sakamuru, J. Zhao, S.A. Shahane, A. Rossoshek, A. Simeonov, Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental

chemicals and drugs, Front. Environ. Sci. 3 (2016), https://doi.org/10.3389/fenvs.2015.00085.

[70] ToxCast. U.S. EPA., 2023.⟨https://www.epa.gov/chemical-research/toxicity-forecaster-toxcasttm-data⟩.

[71] K. Wu, G.-W. Wei, Quantitative toxicity prediction using topology based multitask deep neural networks, J. Chem. Inf. Model. 58 (2) (2018) 520–531, https://doi.org/10.1021/acs.jcim.7b00558.

[72] H. Feng, G.-W. Wei, Virtual screening of drugbank database for herg blockers using topological laplacian-assisted ai models, Comput. Biol. Med. 153 (2023), 106491. ⟨https://www.sciencedirect.com/science/article/pii/S0010482522011994⟩.

[73] J. Jiang, R. Wang, G.-W. Wei, Ggl-tox: geometric graph learning for toxicity prediction, J. Chem. Inf. Model. 61 (4) (2021) 1691–1700, https://doi.org/10.1021/acs.jcim.0c01294.

[74] V. Venkatraman, et al., Drugsniffer: an open source workflow for virtually screening billions of molecules for binding affinity to protein targets, Front. Pharmacol. 13 (2022), https://doi.org/10.3389/fphar.2022.874746.

[75] R.J. Young, Physical Properties in Drug Design, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 1–68, https://doi.org/10.1007/7355_2013_35.

[76] Y. Lu, S. Anand, W. Shirley, P. Gedeck, B.P. Kelley, S. Skolnik, S. Rodde, M. Nguyen, M. Lindvall, W. Jia, Prediction of pka using machine learning methods with rooted topological torsion fingerprints: application to aliphatic amines, J. Chem. Inf. Model. 59 (11) (2019) 4706–4719, https://doi.org/10.1021/acs.jcim.9b00498.

[77] M. Li, H. Zhang, B. Chen, Y. Wu, L. Guan, Prediction of pKa values for neutral and basic drugs based on hybrid artificial intelligence methods, Sci. Rep. 8 (1) (2018), https://doi.org/10.1038/s41598-018-22332-7.

[78] K. Mansouri, et al., Open-source QSAR models for pKa prediction using multiple machine learning approaches, J. Chemin.-. 11 (1) (2019), https://doi.org/10.1186/s13321-019-0384-1.

[79] M. Baltruschat, P. Czodrowski, Machine learning meets pKa, F1000Research 9 (2020) 113, https://doi.org/10.12688/f1000research.22090.2.

[80] P. Hunt, L. Hosseini-Gerami, T. Chrien, J. Plante, D.J. Ponting, M. Segall, Predicting pka using a combination of semi-empirical quantum mechanics and radial basis function methods, J. Chem. Inf. Model. 60 (6) (2020) 2989–2997, https://doi.org/10.1021/acs.jcim.0c00105.

[81] R. Lawler, Y.-H. Liu, N. Majaya, O. Allam, H. Ju, J.Y. Kim, S.S. Jang, Dft-machine learning approach for accurate prediction of pka, J. Phys. Chem. A 125 (39) (2021) 8712–8722, https://doi.org/10.1021/acs.jpca.1c05031.

[82] J. Wu, Y. Wan, Z. Wu, S. Zhang, D. Cao, C.-Y. Hsieh, T. Hou, Mf-sup-pka: multi-fidelity modeling with subgraph pooling mechanism for pka prediction, Acta Pharm. Sin. B 13 (6) (2023) 2572–2584. ⟨https://www.sciencedirect.com/science/article/pii/S2211383522004622⟩.

[83] X. Pan, H. Wang, C. Li, J.Z.H. Zhang, C. Ji, Molgpka: a web server for small molecule pka prediction using a graph-convolutional neural network, J. Chem. Inf. Model. 61 (7) (2021) 3159–3165, https://doi.org/10.1021/acs.jcim.1c00075.

[84] J. Xiong, Z. Li, G. Wang, Z. Fu, F. Zhong, T. Xu, X. Liu, Z. Huang, X. Liu, K. Chen, H. Jiang, M. Zheng, Multi-instance learning of graph neural networks for aqueous pKa prediction, Bioinform 38 (3) (2021) 792–798, https://doi.org/10.1093/bioinformatics/btab714.

[85] R.C. Johnston, K. Yao, Z. Kaplan, M. Chelliah, K. Leswing, S. Seekins, S. Watts, D. Calkins, J. ChiefElk, S.V. Jerome, M.P. Repasky, J.C. Shelley, Epik: pka and protonation state prediction through machine learning, J. Chem. Theory Comp. 19 (8) (2023) 2380–2388, https://doi.org/10.1021/acs.jctc.3c00044.

[86] G. Falcón-Cano, C. Molina, and M.A. Cabrera-Pérez, ADME prediction with KNIME: In silico aqueous solubility models based on supervised recursive machine learning approaches, ADMET and DMPK, 2020.10.5599/admet.852.

[87] K. Liu, et al., Chemi-net: of MolA molecular graph convolutional network for accurate drug property prediction, J. Fan, Chemi-net: Mol. Sci. 20 (14) (2019). ⟨https://www.mdpi.com/1422-0067/20/14/3389⟩.

[88] F. Broccatelli, et al., Benchmarking accuracy and generalizability of four graph neural networks using large in vitro adme datasets from different chemical spaces, Mol. Inf. 41 (8) (2022), 2100321, https://doi.org/10.1002/minf.202100321.

[89] A. Koutsoukas, G. Chang, C.E. Keefer, In-silico extraction of design ideas using mmpa-by-qsar and its application on adme endpoints, J. Chem. Inf. Model. 59 (1) (2019) 477–485, https://doi.org/10.1021/acs.jcim.8b00520.

[90] Z.-M. Win, A.M.Y. Cheong, W.S. Hopkins, Using machine learning to predict partition coefficient (log p) and distribution coefficient (log d) with molecular descriptors and liquid chromatography retention time, J. Chem. Inf. Model. 63 (7) (2023) 1906–1913, https://doi.org/10.1021/acs.jcim.2c01373.

[91] J. Wenzel, H. Matter, F. Schmidt, Predictive multitask deep neural network models for adme-tox properties: learning from large data sets, J. Chem. Inf. Model. 59 (3) (2019) 1253–1268, https://doi.org/10.1021/acs.jcim.8b00785.

[92] M. Petukh, S. Stefl, E. Alexov, The role of protonation states in ligand-receptor recognition and binding, Curr. Pharm. Des. 19 (23) (2013) 4182–4190.

[93] R.E. Carhart, D.H. Smith, R. Venkataraghavan, Atom pairs as molecular features in structure-activity studies: definition and applications, J. Chem. Inf. Comp. Sci. 25 (2) (1985) 64–73, https://doi.org/10.1021/ci00046a002.

[94] R. Nilakantan, N. Bauman, J. Dixon, R. Venkataraghavan, Topological torsion a new molecular descriptor for sar applications comparison with other descriptors, J. Chem. Inf. Comp. Sci. 27 (2) (1987) 82–85. ⟨https://eurekamag.com/research/006/814/006814348.php⟩.

[95] H.L. Morgan, The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service, J. Chem. Doc. 5 (1965) 107–113.

[96] G.R. Bickerton, G.V. Paolini, J. Besnard, S. Muresan, A.L. Hopkins, Quantifying the chemical beauty of drugs, Nat. Chem. 4 (2) (2012) 90–98.

[97] G. Landrum, Rdkit: Open-source cheminformatics software, 2016.⟨https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4⟩.

[98] C. Liao, M.C. Nicklaus, Comparison of nine programs predicting pka values of pharmaceutical substances, J. Chem. Inf. Model. 49 (12) (2009) 2801–2812, https://doi.org/10.1021/ci900289x.

[99] M. Morgenthaler, et al., Predicting and tuning physicochemical properties in lead optimization: amine basicities, ChemMedChem 2 (8) (2007) 1100–1115, https://doi.org/10.1002/cmdc.200700059.

[100] F. Luan, W. Ma, H. Zhang, X. Zhang, M. Liu, Z. Hu, B. Fan, Prediction of pKa for neutral and basic drugs based on radial basis function neural networks and the heuristic method, Pharm. Res. 22 (9) (2005) 1454–1460, https://doi.org/10.1007/s11095-005-6246-8.

[101] C. Dardonville, Automated techniques in pka determination: low, medium and high-throughput screening methods, Drug Disc. Today.: Technol. 27 (2018) 49–58 (physicochemical characterisation in drug discovery), ⟨https://www.sciencedirect.com/science/article/pii/S1740674917300367⟩.

[102] J. Reijenga, et al., Development of methods for the determination of pka values, ACI.S12304, Anal. Chem. Insights 8 (2013), https://doi.org/10.4137/ACI.S12304.

[103] pKa Determination.John Wiley & Sons, Ltd, 2012, ch. 3, 31–173.10.1002/9781118286067.ch3.

[104] M.L. Connolly, Computation of molecular volume, J. Am. Chem. Soc. 107 (5) (1985) 1118–1124, https://doi.org/10.1021/ja00291a006.

[105] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, M. Zheng, Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism, J. Med. Chem. 63 (16) (2020) 8749–8760, https://doi.org/10.1021/acs.jmedchem.9b00959.

[106] J. Kennedy, R. Eberhart, Particle swarm optimization, Proc. ICNN'95 - Int. Conf. Neural Netw. 4 (1995) 1942–1948.

[107] Y. Shi and R. Eberhart, A modified particle swarm optimizer, In: 1998 IEEE Int. Conf. on Evolut. Comp. Proc. IEEE World Congress on Comp. Intell. (Cat. No.98TH8360), 1998, 69–73.

[108] M. Skalic, et al., PlayMolecule BindScope: large scale CNN-based virtual screening on the web, Bioinf 35 (7) (2018) 1237–1238, https://doi.org/10.1093/bioinformatics/bty758.

[109] J. Jiménez, et al., Kdeep: protein-ligand absolute binding affinity prediction via 3d-convolutional neural networks, J. Chem. Inf. Model. 58 (2) (2018) 287–296, https://doi.org/10.1021/acs.jcim.7b00650.

[110] P. Morris, R. Clair St., W.E. Hahn, E. Barenholtz, Predicting binding from screening assays with transformer network embeddings, J. Chem. Inf. Model. 60 (9) (2020) 4191–4199, https://doi.org/10.1021/acs.jcim.9b01212.

[111] W. Torng, R.B. Altman, Graph convolutional neural networks for predicting drug-target interactions, J. Chem. Inf. Model. 59 (10) (2019) 4131–4149, https://doi.org/10.1021/acs.jcim.9b00628.

[112] Y.O. Adeshina, E.J. Deeds, and J. Karanicolas, Machine learning classification can reduce false positives in structure-based virtual screening, Proc. of the Nat. Academy of Sci., 117(31), 18477–18488, 2020.10.1073/pnas.2000585117.

[113] M.S. Nogueira, O. Koch, The development of target-specific machine learning models as scoring functions for docking-based target prediction, J. Chem. Inf. Model. 59 (3) (2019) 1238–1252, https://doi.org/10.1021/acs.jcim.8b00773.

[114] J. Lim, S. Ryu, K. Park, Y.J. Choe, J. Ham, W.Y. Kim, Predicting drug-target interaction using a novel graph neural network with 3d structure-embedded graph representation, J. Chem. Inf. Model. 59 (9) (2019) 3981–3988, https://doi.org/10.1021/acs.jcim.9b00387.

[115] R. Özçelik, et al., Chemboost: A chemical language based approach for protein - ligand binding affinity prediction, Mol. Inf. 40 (5) (2021), 2000212, https://doi.org/10.1002/minf.202000212.

[116] Y. Pu, J. Li, J. Tang, F. Guo, Deepfusiondta: Drug-target binding affinity prediction with information fusion and hybrid deep-learning ensemble model, IEEE/ACM Trans. Comp. Biol. Bioinf. 19 (5) (2022) 2760–2769.

[117] Q. Zhao, F. Xiao, M. Yang, Y. Li, and J. Wang, Attentiondta: prediction of drug-target binding affinity using attention model, In: 2019 IEEE Int. Conf. on Bioinf. and Biomed. (BIBM), 2019, 64–69.

[118] H. Öztürk, A. Özgür, E. Ozkirimli, DeepDTA: deep drug-target binding affinity prediction, Bioinf 34 (17) (2018) i821–i829, https://doi.org/10.1093/bioinformatics/bty593.

[119] J. Shim, Z.-Y. Hong, I. Sohn, C. Hwang, Prediction of drug-target binding affinity using similarity-based convolutional neural network, Sci. Rep. 11 (1) (2021), https://doi.org/10.1038/s41598-021-83679-y.

[120] N. Sánchez-Cruz, J.L. Medina-Franco, J. Mestres, X. Barril, Extended connectivity interaction features: improving binding affinity prediction through chemical description, Bioinf 37 (10) (2020) 1376–1382, https://doi.org/10.1093/bioinformatics/btaa982.

[121] D.D. Wang, H. Xie, H. Yan, Proteo-chemometrics interaction fingerprints of protein-ligand complexes predict binding affinity, Bioinf 37 (17) (2021) 2570–2579, https://doi.org/10.1093/bioinformatics/btab132.

[122] S. Seo, J. Choi, S. Park, J. Ahn, Binding affinity prediction for protein-ligand complex using deep attention mechanism based on intermolecular interactions, BMC Bioinf. 22 (1) (2021), https://doi.org/10.1186/s12859-021-04466-0.

[123] M. Rayka, M.H. Karimi-Jafari, R. Firouzi, Et-score: Improving protein-ligand binding affinity prediction based on distance-weighted interatomic contact

features using extremely randomized trees algorithm, Mol. Inf. 40 (8) (2021), 2060084, https://doi.org/10.1002/minf.202060084.

[124] S. Kumar, M. Hyun Kim, SMPLIP-score: predicting ligand binding affinity from simple and interpretable on-the-fly interaction fingerprint pattern descriptors, J. Chemin.-. 13 (1) (2021), https://doi.org/10.1186/s13321-021-00507-1.

[125] A.D. daSilva, G. Bitencourt-Ferreira, W.F. de Azevedo Jr, Taba: a tool to analyze the binding affinity, J. Comp. Chem. 41 (1) (2020) 69–73, https://doi.org/10.1002/jcc.26048.

[126] F. Zhu, X. Zhang, J.E. Allen, D. Jones, F.C. Lightstone, Binding affinity prediction by pairwise function based on neural network, J. Chem. Inf. Model. 60 (6) (2020) 2766–2772, https://doi.org/10.1021/acs.jcim.0c00026.

[127] L. Zheng, J. Fan, Y. Mu, Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein-ligand binding affinity prediction, ACS Omega 4 (14) (2019) 15956–15965, https://doi.org/10.1021/acsomega.9b01997.

[128] M. Wójcikowski, et al., Development of a protein-ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions, Bioinf 35 (8) (2018) 1334–1341, https://doi.org/10.1093/bioinformatics/bty757.

[129] F. Leidner, N. KurtYilmaz, C.A. Schiffer, Target-specific prediction of ligand affinity with structure-based interaction fingerprints, J. Chem. Inf. Model. 59 (9) (2019) 3679–3691, https://doi.org/10.1021/acs.jcim.9b00457.

[130] E.N. Feinberg, et al., Potentialnet for molecular property prediction, ACS Cent. Sci. 4 (11) (2018) 1520–1530, https://doi.org/10.1021/acscentsci.8b00507.

[131] K. Osaki, T. Ekimoto, T. Yamane, M. Ikeguchi, 3d-rism-ai: a machine learning approach to predict protein-ligand binding affinity using 3d-rism, J. Phys. Chem. B 126 (33) (2022) 6148–6158, https://doi.org/10.1021/acs.jpcb.2c03384.

[132] J. Lu, X. Hou, C. Wang, Y. Zhang, Incorporating explicit water molecules and ligand conformation stability in machine-learning scoring functions, J. Chem. Inf. Model. 59 (11) (2019) 4540–4549, https://doi.org/10.1021/acs.jcim.9b00645.

[133] L. Dong, X. Qu, Y. Zhao, B. Wang, Prediction of binding free energy of protein-ligand complexes with a hybrid molecular mechanics/generalized born surface area and machine learning method, ACS Omega 6 (48) (2021) 32938–32947, https://doi.org/10.1021/acsomega.1c04996.

[134] F. Boyles, C.M. Deane, G.M. Morris, Learning from the ligand: using ligand-based features to improve binding affinity prediction, Bioinf 36 (3) (2019) 758–764, https://doi.org/10.1093/bioinformatics/btz665.

[135] K.J. Fujimoto, S. Minami, T. Yanai, Machine-learning- and knowledge-based scoring functions incorporating ligand and protein fingerprints, ACS Omega 7 (22) (2022) 19030–19039, https://doi.org/10.1021/acsomega.2c02822.

[136] S. Holderbach, L. Adam, B. Jayaram, R.C. Wade, G. Mukherjee, Raspd.: fast protein-ligand binding free energy prediction using simplified physicochemical features, Front. Mol. Biosci. 7 (2020), https://doi.org/10.3389/fmolb.2020.601065.

[137] Z. Meng, K. Xia, Persistent spectral-based machine learning (PerSpect ML) for protein-ligand binding affinity prediction, Sci. Adv. 7 (19) (2021), https://doi.org/10.1126/sciadv.abc5329.

[138] J. Wee, K. Xia, Forman persistent Ricci curvature (FPRC)-based machine learning models for protein-ligand binding affinity prediction, bbab136, Brief. Bioinf. 22 (6) (2021), https://doi.org/10.1093/bib/bbab136. bbab136.

[139] D.D. Nguyen, et al., Mathematical deep learning for pose and binding affinity prediction and ranking in d3r grand challenges, J. Comput. -Aided Mol. Des. 33 (1) (2018) 71–82, https://doi.org/10.1007/s10822-018-0146-6.

[140] D.D. Nguyen, G.-W. Wei, Agl-score: Algebraic graph learning score for protein-ligand binding scoring, ranking, docking, and screening, J. Chem. Inf. Model. 59 (7) (2019) 3291–3304, https://doi.org/10.1021/acs.jcim.9b00334.

[141] R. Liu, X. Liu, J. Wu, Persistent path-spectral (pps) based machine learning for protein-ligand binding affinity prediction, J. Chem. Inf. Model. 63 (3) (2023) 1066–1075, https://doi.org/10.1021/acs.jcim.2c01251.

[142] M.M. Rana, D.D. Nguyen, Geometric graph learning with extended atom-types features for protein-ligand binding affinity prediction, Comput. Biol. Med. 164 (2023), 107250. ⟨https://www.sciencedirect.com/science/article/pii/S0010482523007151⟩.

[143] M.A. Rezaei, Y. Li, D. Wu, X. Li, C. Li, Deep learning in drug design: protein-ligand binding affinity prediction, IEEE/ACM Trans. Comp. Biol. Bioinf. 19 (1) (2022) 407–417.

[144] M.M. Stepniewska-Dziubinska, P. Zielenkiewicz, P. Siedlecki, Development and evaluation of a deep learning model for protein-ligand binding affinity prediction, Bioinf 34 (21) (2018) 3666–3674, https://doi.org/10.1093/bioinformatics/bty374.

[145] P.G. Francoeur, et al., Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design, J. Chem. Inf. Model. 60 (9) (2020) 4200–4215, https://doi.org/10.1021/acs.jcim.0c00411.

[146] Y. Kwon, et al., Ak-score: of Mol Accurate protein-ligand binding affinity prediction using an ensemble of 3d-convolutional neural networks, J. Lee, Ak-score: Mol. Sci. 21 (22) (2020). ⟨https://www.mdpi.com/1422-0067/21/22/8424⟩.

[147] R. Meli, et al., Learning protein-ligand binding affinity with atomic environment vectors, J. Chemin.-. 13 (1) (2021), https://doi.org/10.1186/s13321-021-00536-w.

[148] H. Yuan, et al., Protein-ligand binding affinity prediction model based on graph attention network, Math. Biosci. Eng. 18 (6) (2021) 9148–9162, https://doi.org/10.3934/mbe.2021451.

[149] X. Qu, et al., Water network-augmented two-state model for protein-ligand binding affinity prediction, 0(0), null, J. Chem. Inf. Model. (2023), https://doi.org/10.1021/acs.jcim.3c00567, 0(0), null.

[150] J. Jiménez-Luna, et al., DeltaDelta neural networks for lead optimization of small molecule potency, Chem. Sci. 10 (47) (2019) 10911–10918, https://doi.org/10.1039/c9sc04606b.

[151] F. Gusev, E. Gutkin, M.G. Kurnikova, O. Isayev, Active learning guided drug design lead optimization based on relative binding free energy modeling, J. Chem. Inf. Model. 63 (2) (2023) 583–594, https://doi.org/10.1021/acs.jcim.2c01052.

[152] R.F. Alford, et al., The rosetta all-atom energy function for macromolecular modeling and design, J. Chem. Theory Comp. 13 (6) (2017) 3031–3048, https://doi.org/10.1021/acs.jctc.7b00125.

[153] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (3) (1990) 403–410. ⟨https://www.sciencedirect.com/science/article/pii/S0022283605803602⟩.

[154] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inf. Model. 50 (5) (2010) 742–754, https://doi.org/10.1021/ci100050t.

[155] J.L. Durant, B.A. Leland, D.R. Henry, J.G. Nourse, Reoptimization of mdl keys for use in drug discovery, J. Chem. Inf. Comp. Sci. 42 (6) (2002) 1273–1280, https://doi.org/10.1021/ci010132r.

[156] D.D. Wang, M.-T. Chan, H. Yan, Structure-based protein-ligand interaction fingerprints for binding affinity prediction, Comp. Struct. Biotech. J. 19 (2021) 6291–6300. ⟨https://www.sciencedirect.com/science/article/pii/S2001037021004839⟩.

[157] Z. Yin, W. Song, B. Li, F. Wang, L. Xie, X. Xu, Neural networks prediction of the protein-ligand binding affinity with circular fingerprints, Tech. Health Care 31 (2023) 487–495, https://doi.org/10.3233/thc-236042.

[158] L. Vietoris, Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen, Math. Ann. 97 (1) (1927) 454–472, https://doi.org/10.1007/bf01447877.

[159] J.S. Smith, O. Isayev, A.E. Roitberg, ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost, Chem. Sci. 8 (4) (2017) 3192–3203, 10.1039-2Fc6sc05720a.

[160] Y. Li, Y. Gao, M.K. Holloway, R. Wang, Prediction of the favorable hydration sites in a protein binding pocket and its application to scoring function formulation, J. Chem. Inf. Model. 60 (9) (2020) 4359–4375, https://doi.org/10.1021/acs.jcim.9b00619.

[161] S. Gu, C. Shen, J. Yu, H. Zhao, H. Liu, L. Liu, R. Sheng, L. Xu, Z. Wang, T. Hou, Y. Kang, Can molecular dynamics simulations improve predictions of protein-ligand binding affinity with machine learning?, bbad008, Brief. Bioinf. 24 (2) (2023), https://doi.org/10.1093/bib/bbad008. bbad008.

[162] T. Cheng, X. Li, Y. Li, Z. Liu, R. Wang, Comparative assessment of scoring functions on a diverse test set, J. Chem. Inf. Model. 49 (4) (2009) 1079–1093, https://doi.org/10.1021/ci9000053.

[163] Y. Li, Z. Liu, J. Li, L. Han, J. Liu, Z. Zhao, R. Wang, Comparative assessment of scoring functions on an updated benchmark: 1. compilation of the test set, J. Chem. Inf. Model. 54 (6) (2014) 1700–1716, https://doi.org/10.1021/ci500080q.

[164] Y. Li, L. Han, Z. Liu, R. Wang, Comparative assessment of scoring functions on an updated benchmark: 2. evaluation methods and general results, J. Chem. Inf. Model. 54 (6) (2014) 1717–1736, https://doi.org/10.1021/ci500081m.

[165] M. Su, et al., Comparative assessment of scoring functions: the casf-2016 update, J. Chem. Inf. Model. 59 (2) (2019) 895–913, https://doi.org/10.1021/acs.jcim.8b00545.

[166] M.J. Hartshorn, et al., Diverse, high-quality test set for the validation of protein-ligand docking performance, J. Med. Chem. 50 (4) (2007) 726–741, https://doi.org/10.1021/jm061277y.

[167] R.D. Smith, et al., Csar benchmark exercise of 2010: combined evaluation across all submitted scoring functions, J. Chem. Inf. Model. 51 (9) (2011) 2115–2131, https://doi.org/10.1021/ci200269q.

[168] J.B.J. Dunbar, et al., Csar benchmark exercise of 2010: selection of the protein-ligand complexes, J. Chem. Inf. Model. 51 (9) (2011) 2036–2046, https://doi.org/10.1021/ci200082t.

[169] K.L. Damm-Ganamet, et al., Csar benchmark exercise 2011-2012: evaluation of results from docking and relative ranking of blinded congeneric series, J. Chem. Inf. Model. 53 (8) (2013) 1853–1870, https://doi.org/10.1021/ci400025f.

[170] J.B.J. Dunbar, et al., Csar data set release 2012: ligands, affinities, complexes, and docking decoys, J. Chem. Inf. Model. 53 (8) (2013) 1842–1852, https://doi.org/10.1021/ci4000486.

[171] R.D. Smith, et al., Csar benchmark exercise 2013: evaluation of results from a combined computational protein design, docking, and scoring/ranking challenge, J. Chem. Inf. Model. 56 (6) (2016) 1022–1031, https://doi.org/10.1021/acs.jcim.5b00387.

[172] H.A. Carlson, et al., Csar 2014: a benchmark exercise using unpublished data from pharma, J. Chem. Inf. Model. 56 (6) (2016) 1063–1077, https://doi.org/10.1021/acs.jcim.5b00523.

[173] G.A. Landrum, M. Beckers, J. Lanini, N. Schneider, N. Stiefl, and S. Riniker, SIMPD: an algorithm for generating simulated time splits for validating machine learning approaches, 2023.10.26434/chemrxiv-2023-x9pjf.

[174] L. Wang, et al., Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field, J. Am. Chem. Soc. 137 (7) (2015) 2695–2703, https://doi.org/10.1021/ja512751q.

[175] D.F. Hahn, et al., Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks [article v1.0], Living J. Comp. Mol. Sci. 4 (1) (2022), https://doi.org/10.33011/livecoms.4.1.1497.

[176] C.E.M. Schindler, et al., Large-scale assessment of binding free energy calculations in active drug discovery projects, J. Chem. Inf. Model. 60 (11) (2020) 5457–5474, https://doi.org/10.1021/acs.jcim.0c00900.

[177] C.D. Parks, et al., D3r grand challenge 4: blind prediction of protein-ligand poses, affinity rankings, and relative binding free energies, J. Comput. -Aided Mol. Des. 34 (2) (2020) 99–119, https://doi.org/10.1007/s10822-020-00289-y.

[178] M. Amezcua, et al., An overview of the SAMPL8 host-guest binding challenge, J. Comput. -Aided Mol. Des. 36 (10) (2022) 707–734, https://doi.org/10.1007/s10822-022-00462-5.

[179] M. Li, S. Xu, X. Cai, Z. Zhang, and H. Ji, Contrastive meta-learning for drug-target binding affinity prediction, In: 2022 IEEE Int. Conf. on Bioinform. and Biomed. (BIBM), 2022, 464–470.

[180] J. Wang, L. Urban, The impact of early adme profiling on drug discovery and development strategy, Drug Discov. World 5 (2004) 73–86.

[181] G.M. Currie, Pharmacology, part 2: introduction to pharmacokinetics, J. Nucl. Med. Tech. 46 (3) (2018) 221–230. ⟨https://tech.snmjournals.org/content/46/3/221⟩.

[182] M.-L. Chen, L. Lesko, R.L. Williams, Measures of exposure versus measures of rate and extent of absorption, Clin. Pharmacokinet. 40 (8) (2001) 565–572, https://doi.org/10.2165/00003088-200140080-00001.

[183] I.D. Angelis, L. Turco, Caco-2 cells as a model for intestinal absorption, Curr. Protoc. Toxicol. 47 (1) (2011), https://doi.org/10.1002/0471140856.tx2006s47.

[184] S. He, A. Zhiti, A. Barba-Bon, A. Hennig, W.M. Nau, Real-time parallel artificial membrane permeability assay based on supramolecular fluorescent artificial receptors, Front. Chem. 8 (2020), https://doi.org/10.3389/fchem.2020.597927.

[185] V.E. Thiel-Demby, et al., Biopharmaceutics classification system: validation and learnings of an in vitro permeability assay, Mol. Pharm. 6 (1) (2009) 11–18, https://doi.org/10.1021/mp800122b.

[186] F. Sharom, The p-glycoprotein efflux pump: how does it transport drugs? J. Membr. Biol. 160 (3) (1997) 161–175, https://doi.org/10.1007/s002329900305.

[187] F. Chaubet, et al., Pharmacology: drug delivery. Encycl. of Biomed. Eng., R. Narayan, Ed, Elsevier,, Oxford, 2019, pp. 440–453. ⟨https://www.sciencedirect.com/science/article/pii/B9780128012383110074⟩.

[188] J. Bernacki, et al., Physiology and pharmacological role of the blood-brain barrier, Pharmacol. Rep.: PR 60 (2007) 600–622.

[189] M. Zhao, et al., Cytochrome p450 enzymes and drug metabolism in humans, Int. J. Mol. Sci. 22 (23) (2021) 12808, https://doi.org/10.3390/ijms222312808.

[190] Y. Parmentier, et al., In vitro studies of drug metabolism. Compr. Med. Chem. II, Elsevier, 2007, pp. 231–257, https://doi.org/10.1016/b0-08-045044-x/00125-5.

[191] X. Ma, J.R. Idle, F.J. Gonzalez, The pregnane x receptor: from bench to bedside, Expert Opin. Drug Metabol. Tox. 4 (7) (2008) 895–908, https://doi.org/10.1517/17425255.4.7.895.

[192] H. Satsu, et al., Activation of pregnane x receptor and induction of MDR1 by dietary phytochemicals, J. Agric. Food Chem. 56 (13) (2008) 5366–5373, https://doi.org/10.1021/jf073350e.

[193] S.A. Kliewer, B. Goodwin, T.M. Willson, The nuclear pregnane X receptor: a key regulator of xenobiotic metabolism, Endocr. Rev. 23 (5) (2002) 687–702, https://doi.org/10.1210/er.2001-0038.

[194] V.K. Bhosle, et al., 18 - basic pharmacologic principles, in: R.A. Polin, S. H. Abman, D.H. Rowitch, W.E. Benitz, W.W. Fox (Eds.), Fetal and Neonatal Physiol. (Fifth Edition), fifth edition, Elsevier, 2017, pp. 187–201.e3. ⟨https://www.sciencedirect.com/science/article/pii/B9780323352147000184⟩.

[195] F.P. Guengerich, Mechanisms of drug toxicity and relevance to pharmaceutical development, Drug Metabol. Pharmacokinet. 26 (1) (2011) 3–14, https://doi.org/10.2133/dmpk.dmpk-10-rv-062.

[196] A. Garrido, et al., hERG toxicity assessment: useful guidelines for drug design, Eur. J. Med. Chem. 195 (2020), 112290, https://doi.org/10.1016/j.ejmech.2020.112290.

[197] L. Meunier, D. Larrey, Drug-induced liver injury: biomarkers, requirements, candidates, and validation, Front. Pharmacol. 10 (2019), https://doi.org/10.3389/fphar.2019.01482.

[198] W. Föllmann, G. Degen, F. Oesch, J. Hengstler, Ames Test, in Brenneras Encycl. of Genet, Elsevier, 2013, pp. 104–107, https://doi.org/10.1016/b978-0-12-374984-0.00048-6.

[199] M. Hayashi, The micronucleus test–most widely used in vivo genotoxicity test– - Genes and Environment — doi.org, 10.1186/s41021–016-0044-x, 2016, [Accessed 16-Jul-2023].

[200] V. Siramshetty, et al., Validating ADME QSAR models using marketed drugs, SLAS Disc. 26 (10) (2021) 1326–1336, https://doi.org/10.1177/24725552211017520.

[201] L. Zhu, et al., ADME properties evaluation in drug discovery: in silico prediction of blood-brain partitioning, Mol. Divers. 22 (4) (2018) 979–990, https://doi.org/10.1007/s11030-018-9866-8.

[202] Y. Zhou, et al., Exploring tunable hyperparameters for deep neural networks with industrial adme data sets, J. Chem. Inf. Model. 59 (3) (2019) 1005–1016, https://doi.org/10.1021/acs.jcim.8b00671.

[203] Y. Kosugi, N. Hosea, Prediction of oral pharmacokinetics using a combination of in silico descriptors and in vitro adme properties, Mol. Pharm. 18 (3) (2021) 1071–1079, https://doi.org/10.1021/acs.molpharmaceut.0c01009.

[204] O. Obrezanova, et al., Prediction of in vivo pharmacokinetic parameters and time-exposure curves in rats using machine learning from the chemical structure, Mol. Pharm. 19 (5) (2022) 1488–1504, https://doi.org/10.1021/acs.molpharmaceut.2c00027.

[205] Y. Kosugi, N. Hosea, Direct comparison of total clearance prediction: Computational machine learning model versus bottom-up approach using in vitro assay, Mol. Pharm. 17 (7) (2020) 2299–2309, https://doi.org/10.1021/acs.molpharmaceut.9b01294.

[206] Y. Yuan, et al., A novel strategy for prediction of human plasma protein binding using machine learning techniques, Chemom. Intell. Lab. Syst. 199 (2020), 103962. ⟨https://www.sciencedirect.com/science/article/pii/S016974391930468X⟩.

[207] F. Miljkovič, et al., Machine learning models for human in vivo pharmacokinetic parameters with in-house validation, Mol. Pharm. 18 (12) (2021) 4520–4530, https://doi.org/10.1021/acs.molpharmaceut.1c00718.

[208] M.A. Lim, et al., Exploring deep learning of quantum chemical properties for absorption, distribution, metabolism, and excretion predictions, J. Chem. Inf. Model. 62 (24) (2022) 6336–6341, https://doi.org/10.1021/acs.jcim.2c00245.

[209] J. Jiang, et al., Boosting tree-assisted multitask deep learning for small scientific datasets, J. Chem. Inf. Model. 60 (3) (2020) 1235–1244, https://doi.org/10.1021/acs.jcim.9b01184.

[210] X. Li, et al., Prediction of admet properties of anti-breast cancer compounds using three machine learning algorithms, Mol 28 (5) (2023). ⟨https://www.mdpi.com/1420-3049/28/5/2326⟩.

[211] Z. Fan, S. Wang, Z. Xie, and Z. Li, Adme prediction for breast cancer drugs in computer-aided drug design, In: Proc. of the 11th Int. Conf. on Inf., Environ., Energy and Appl., ser. IEEA '22. Association for Computing Machinery, 2022, 14–18.10.1145/3533254.3533257.

[212] G. Falcón-Cano, C. Molina, M.n. Cabrera-Pérez, Adme prediction with knime: development and validation of a publicly available workflow for the prediction of human oral bioavailability, J. Chem. Inf. Model. 60 (6) (2020) 2660–2667, https://doi.org/10.1021/acs.jcim.0c00019.

[213] Y. Chen, et al., In silico prediction of herg blockers using machine learning and deep learning approaches (n/a(n/a)), J. Appl. Tox. (2023), https://doi.org/10.1002/jat.4477 (n/a(n/a)).

[214] A. Orosz, K. Héberger, A. Rácz, Comparison of descriptor- and fingerprint sets in machine learning models for adme-tox targets, Front. Chem. 10 (2022), https://doi.org/10.3389/fchem.2022.852893.

[215] M. Yang, et al., A novel adaptive ensemble classification framework for adme prediction, RSC Adv. 8 (2018) 11661–11683, https://doi.org/10.1039/C8RA01206G.

[216] M. Kursa, A. Jankowski, W. Rudnicki, Boruta - a system for feature selection, Fundam. Inform. 101 (2010) 271–285.

[217] A.M. Doweyko, 3d-QSAR illusions, J. Comput. -Aided Mol. Des. 18 (7–9) (2004) 587–596, https://doi.org/10.1007/s10822-004-4068-0.

[218] B. Sanchez-Lengeling, et al., Evaluating attribution for graph neural networks, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 33. Curran Associates, Inc., 5898–5910, 2020.⟨https://proceedings.neurips.cc/paper_files/paper/2020/file/417fbbf2e9d5a28a855a11894b2e795a-Paper.pdf⟩.

[219] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Adv. Large Margin Classif. 10 (2000).

[220] A. Saabas, Interpreting random forests, Diving Into Data 24 (2014).

[221] S.M. Lundberg, et al., From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (1) (2020) 56–67, https://doi.org/10.1038/s42256-019-0138-9.

[222] S. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, 2017.

[223] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, Visualizing higher-layer features of a deep network, Technical Report, Univeristé de Montréal, 2009.

[224] L. McInnes, J. Healy, and J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[225] S. Carter, Exploring neural networks with activation atlases, 2019.

[226] K. Simonyan, A. Vedaldi, and A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[227] T. Schnake, et al., Higher-order explanations of graph neural networks via relevant walks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2022) 7581–7596, 10.1109-2Ftpami.2021.3115452.

[228] G.P. Wellawatte, A. Seshadri, A.D. White, Model agnostic generation of counterfactual explanations for molecules, Chem. Sci. 13 (13) (2022) 3697–3705, https://doi.org/10.1039/d1sc05259d.