Received: 8 April 2019 | Revised: 13 May 2019 | Accepted: 15 May 2019

DOI: 10.1002/wcms.1429

ADVANCED REVIEW

WIREs COMPUTATIONAL MOLECULAR SCIENCE WILEY

From machine learning to deep learning: Advances in scoring functions for protein–ligand docking

Chao Shen[1] | Junjie Ding[2] | Zhe Wang[1] | Dongsheng Cao[3] | Xiaoqin Ding[2] | Tingjun Hou[1]

[1]Hangzhou Institute of Innovative Medicine, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, P. R. China

[2]Beijing Institute of Pharmaceutical Chemistry, Beijing, P. R. China

[3]Xiangya School of Pharmaceutical Sciences, Central South University, Changsha, P. R. China

Correspondence
Tingjun Hou, Hangzhou Institute of Innovative Medicine, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, Zhejiang 310058, P. R. China.
Email: tingjunhou@zju.edu.cn

Funding information
National Key R&D Program of China, Grant/Award Numbers: 2016YFA0501701, 2016YFB0201700; National Natural Science Foundation of China, Grant/Award Numbers: 21575128, 81603031, 81773632; Zhejiang Provincial Natural Science Foundation of China, Grant/Award Number: LZ19H300001

Abstract

Molecule docking has been regarded as a routine tool for drug discovery, but its accuracy highly depends on the reliability of scoring functions (SFs). With the rapid development of machine learning (ML) techniques, ML-based SFs have gradually emerged as a promising alternative for protein–ligand binding affinity prediction and virtual screening, and most of them have shown significantly better performance than a wide range of classical SFs. Emergence of more data-hungry deep learning (DL) approaches in recent years further fascinates the exploitation of more accurate SFs. Here, we summarize the progress of traditional ML-based SFs in the last few years and provide insights into recently developed DL-based SFs. We believe that the continuous improvement in ML-based SFs can surely guide the early-stage drug design and accelerate the discovery of new drugs.

This article is categorized under:

Computer and Information Science > Chemoinformatics

KEYWORDS

deep learning, machine learning, molecular docking, scoring function, structure-based drug design

1 | INTRODUCTION

Traditional drug discovery largely relies on the application of high-throughput screening, an experimental technique with acceptable performance but high cost and low efficiency.[1] With the rapid development of computational chemistry and computer technology, computer-aided drug design (CADD) has gradually emerged as a powerful technique in the design and development of new drug candidates in the past three decades.[2] Virtual screening (VS), an important branch of CADD, can enrich potential actives from large virtual compound libraries through in silico methods rather than real experiments, which can not only accelerate the process of drug discovery but also greatly reduce the time and resource cost.[3–5] Depending on whether the three-dimensional (3D) structure of a target is used or not, VS approaches can be classified into two major categories: ligand-based virtual screening (LBVS) and structure-based virtual screening (SBVS).[6] LBVS aims to discover active molecules through the models developed based on a set of known ligands of a target of interest, which may limit its capability to find novel chemotypes. Compared with LBVS, SBVS is considered to be a better choice to discover novel active compounds if the 3D structure of a given target is available.[7]

Chao Shen and Junjie Ding are equivalent first authors.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

© 2019 Wiley Periodicals, Inc.

https://doi.org/10.1002/wcms.1429

https://doi.org/10.1002/wcms.1429

As one of the most frequently used approaches in SBVS, molecular docking can not only predict the binding conformation of a ligand in the binding pocket of the target but also estimate the binding affinity of a ligand to its target.[8] The reliability of docking depends on the coverage of the conformational sampling and the quality of the scoring function (SF). The role of the conformational sampling is to search poses within a given conformational space, whereas the SF gives each pose a score to represent its relative binding affinity, and then the top-ranked pose is regarded as the final docking pose.[9] Although molecular docking is computationally efficient, the binding affinities estimated by SFs in many cases do not correlate well with experimentally determined binding affinities, and even cannot effectively distinguish actives from non-actives.[10] Over the past two decades, extensive efforts have been dedicated to improve the existing SFs or developing innovative ones, but nowadays, pursuing a highly applicable SF is still quite challenging in the context of computational chemistry.[11]

Machine learning (ML) approaches, which use pattern recognition algorithms to discern mathematical relationships between empirical observations, have been widely applied in the field of biomedicine, such as medical image classification,[12,13] protein secondary structure prediction,[14,15] drug repositioning,[16,17] drug design,[18–20] etc. In recent years, a more data-hungry ML algorithm, deep learning (DL), which has gained great success in a wide variety of applications, such as computer vision,[21,22] speech recognition,[23] computer games,[24] and natural language processing,[25] has also attracted considerable interest from computational chemists and medicinal chemists. Up to now, various reviews related to the applications of ML or DL in drug design and discovery have been published.[18–20,26–34] Ain et al.[35] and Khamis et al.[36] summarized the advances of ML-based SFs before 2015 in two comprehensive reviews about protein–ligand binding affinity prediction and SBVS, but DL has just begun to rise in the field of drug discovery in 2015.[29] Hence, in this study, in addition to reviewing the advances of ML-based SFs in recent years, we mainly focus on the difference between traditional ML-based SFs and DL-based SFs. It is expected that our study can provide some valuable guidance for the exploitation and development of more reliable ML (DL)-based SFs.

2 | OVERVIEW OF SCORING FUNCTIONS

Classical SFs can be typically divided into three classes, namely force field (FF)-based SFs, empirical SFs and knowledge-based SFs (Figure 1).[37] In general, an FF-based SF can always be described as a weighted sum of several non-bonded (such as van der Waals, electrostatic and hydrogen bonding) interaction terms computed by a given FF, and the weights for all terms are set to 1 by default. To enhance computational efficiency, some hard-handled terms such as desolvation and entropic contributions are often simplified or even neglected in FF-based SFs. As a result, these SFs may not be able to accurately rank the

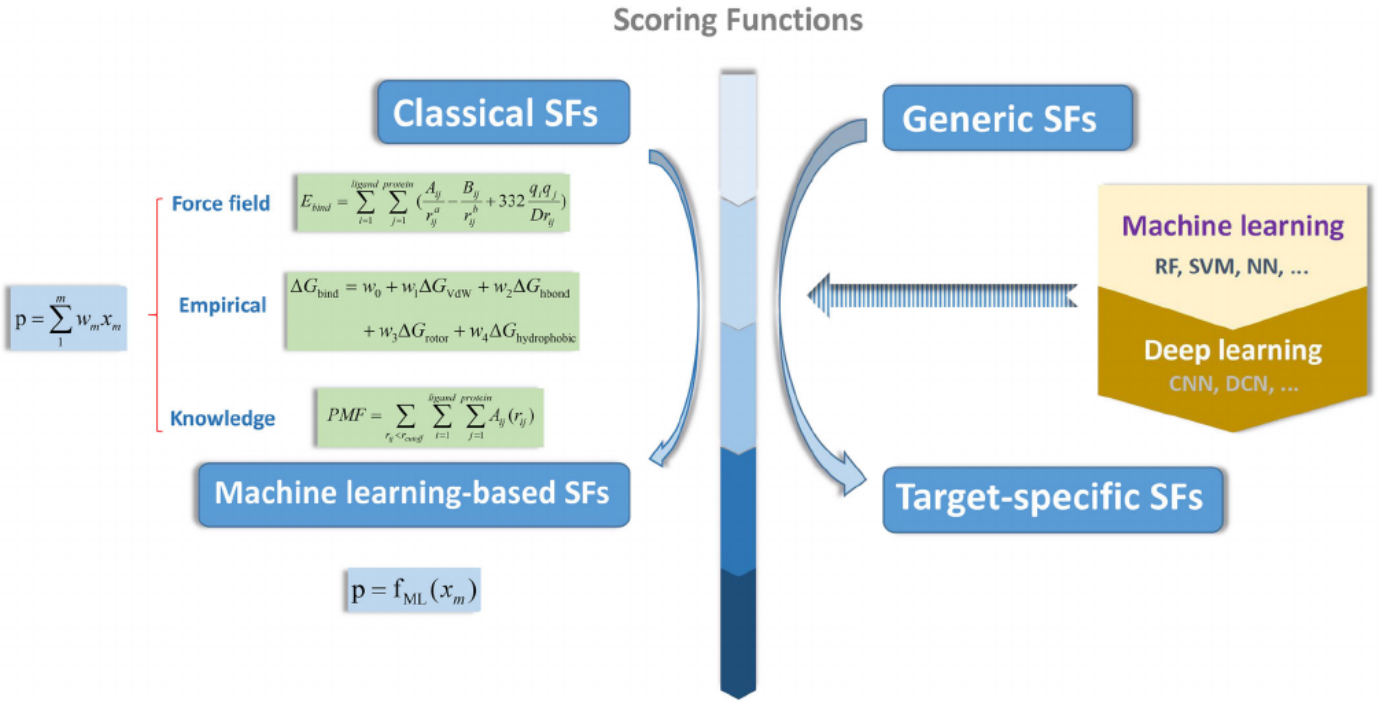
Scoring Functions
Classical SFs
Generic SFs
Force field
$E_{bind} = \sum_{i=1}^{ligand} \sum_{j=1}^{protein} (\frac{A_{ij}}{r_{ij}^{a}} - \frac{B_{ij}}{r_{ij}^{b}} + 332\frac{q_i q_j}{Dr_{ij}})$
Machine learning
RF, SVM, NN, ...
$p = \sum_{1}^{m} w_m x_m$
Empirical
$\Delta G_{\text{bind}} = w_0 + w_1\Delta G_{\text{VdW}} + w_2\Delta G_{\text{hbond}} + w_3\Delta G_{\text{rotor}} + w_4\Delta G_{\text{hydrophobic}}$
Deep learning
CNN, DCN, ...
Knowledge
$PMF = \sum_{r_{ij}<r_{cutoff}} \sum_{i=1}^{ligand} \sum_{j=1}^{protein} A_{ij}(r_{ij})$
Machine learning-based SFs
Target-specific SFs
$p = f_{ML}(x_m)$

FIGURE 1 Overview of scoring functions

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

protein–ligand binding affinities or identify true binders for some drug targets.[38] FF-based SFs have been implemented in many commonly used docking programs, such as DOCK,[39] GOLD,[40] and LigandFit.[41] Empirical SFs adopt similar functional forms used by FF-based SFs except that some additional terms, such as the counts of the rotatable bonds of the ligands, solvent accessibility surface area (SASA), etc., are added to provide better characterization of protein–ligand interactions. Besides, unlike the identical weights for all terms in an FF-based SF, each term in an empirical SF has its own weight, which can be obtained by linearly fitting the scoring terms to experimental binding affinities.[42] Empirical SFs account for the largest proportion of all the SFs, and many well-known SFs belong to this class, such as Autodock Vina,[43] GlideScore,[44,45] ChemScore,[46] and X-Score.[47] Compared with FF-based SFs and empirical SFs, knowledge-based SFs can be considered as a group of inherently different methods. In these SFs, the 3D coordinates of a large set of protein–ligand complexes are regarded as a knowledge base, and then the relative occurrence frequencies of some features, such as atom–atom pairwise contacts, are calculated.[48] Popular implementations of knowledge-based SFs include PMF,[49] DrugScore,[50] and ASP.[51]

A common feature of the above three types of classical SFs is that they all assume an additive functional form to represent the linear relationship between the binding affinity and the features that characterize a protein–ligand complex. But actually, such a linear relationship may not always exist. In other words, if a nonlinear method is used to fit this relationship, the obtained SF may have a better performance.[52] Therefore, some ML methods, such as random forest (RF),[53] support vector machine (SVM),[54,55] and neural network (NN),[56] have been used to replace traditional multiple linear regression (MLR) or partial least squares method in the development of SFs, thereby leading to the appearance of the concept of "machine learning-based SFs."

Another classification can divide SFs into generic and target-specific (or family-specific) SFs. Just as the names imply, the former is designed to be applicable to all targets, whereas the latter just applicable to a specific target or a target family. Due to the fact that most generic SFs cannot yield ideal performances in all situations, more target-specific ones are successively developed when performing a SBVS towards a certain target especially with the involvement of a lot of advanced ML technologies.[57] In this study, as we mainly focus on the different types of ML methods, generic and target-specific SFs are not specially distinguished.

3 | BENCHMARKS TO CONSTRUCT OR ASSESS A SCORING FUNCTION

When developing a SF, different benchmark datasets will be used with different purposes. Table 1 lists several widely used datasets and their compositions. Among all the datasets, PDBbind database,[58–64] which provides a consolidated repository of the bioactivity data (K_d or K_i) for all types of biomolecular complexes extracted from the Protein Data Bank (PDB),[65] is the most common dataset used to develop SFs for binding affinity prediction. It is updated annually and the newest version v2018 contains a total of 19,588 protein–ligand complexes, of which the 4,463 ones with high quality are named as the refined set and the 285 ones are further selected as the core set which is always used for the assessment of SFs. Directory of Useful Decoys (DUD),[66] Database of Useful Decoys-Enhanced (DUD-E),[67] and Maximum Unbiased Validation (MUV)[68]

TABLE 1 Summary of some commonly used datasets

Dataset		Contained complexes (or compounds)		
	Version[a]	General set	Refined set	Core set
PDBbind[58–64a]	v2002	1,446	800	—
	v2007	3,124	1,300	195
	v2009	5,678	1,741	219
	v2013	10,776	2,959	195
	v2015	14,260	3,706	195
	v2016	16,179	4,057	285
	v2018	19,588	4,463	285
DUD[66]		40 targets, 2,950 active compounds, and 36 decoys for each active compound		
DUD-E[67]		102 targets, 22,886 active compounds, and 50 decoys for each active compound		
MUV[68]		17 targets, 510 active compounds, and 50 decoys for each active compound		

[a] Only the most commonly used versions of PDBbind dataset are listed.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

benchmarks that incorporate decoys into the datasets, however, are specially designed to develop SFs for VS. In each dataset, several representative targets and their corresponding active compounds are collected and then 50 decoys (36 for the DUD) that have similar physicochemical properties but dissimilar two-dimensional topology are generated for each active compound so that not only the numbers of active molecules and background decoys can be surely biased but the effects caused by some ligand properties such as molecule weight can also be reduced to the minimum. For developing a target-specific SF, the experimental dataset can be retrieved from some websites such as BindingDB,[69] PubChem,[70] and ChEMBL.[71] In addition, the above datasets can also be used to assess the performance of a SF. However, most of them are clearly published, suggesting that the known binding data used in developing SFs may also be used in assessing SFs. Hence, some blinded prediction tests based on unpublished datasets are organized to evaluate the prediction capability of SFs in a more fair way, such as the Community Structure–Activity Resource (CSAR) exercises[72–74] and the D3R Grand Challenge.[75,76]

As for the assessment of SFs, the most popular benchmark is the one named CASF first proposed by Cheng et al. in 2009,[59] and now the newest version has come up to v2016.[64] The CASF benchmark evaluate a SF based on the following four criteria: the scoring power, the ranking power, the docking power and the screening power, and all the assessments are based on the PDBbind core set. The scoring power refers to the ability of a SF to produce a linear correlation between the calculated binding affinities and experimental binding data. The benchmark dataset used by CASF-2016 contains 285 high quality protein–ligand complexes, and the classic Pearson's correlation coefficient (R_p) and the SD in regression are employed to evaluate the scoring power of a SF. In addition, other metrics such as root mean square error and Spearman's correlation coefficient (R_s) are also commonly used.[35] The ranking power is used to evaluate whether a SF can correctly rank the known ligands of the same target. The test set consists of 57 clusters of complexes, each of which has five complexes with significantly different binding affinities. The Spearman's correlation coefficient (R_s), Kendall's correlation coefficient (R_k), and Predictive Index are served as the quantitative indicators of the ranking power. The docking power refers to the capacity of a SF to distinguish near-native poses from computer-generated decoys. Traditionally, the near-native poses are mixed with the decoy poses, and if the root mean square deviation (RMSD) between the top-ranked pose and the experimentally observed (native) binding pose is less than a predefined cutoff (e.g., 2.0 Å), the prediction is considered successful. The overall success rate for the test set is used to represent the final docking power. The screening power is used to evaluate whether a SF can identify true binders from a set of random molecules. The assessment in CASF is based on the above 285 complexes, in which the five ligands towards a certain target are the actives while the other 280 ligands are considered as the inactives. The enrichment factors (EFs) calculated by counting the total number of the true binders among the 1, 5, and 10% top-ranked molecules are utilized to quantify the screening power. However, different from the CASF, the evaluations of the screening power in some studies are just based on the true binders and their corresponding computer-generated decoys. And besides the EFs, the area under the receiver operating characteristic curve (AUC) and the area under the precision versus recall curve (AUPR) are also widely used for this purpose. In general, EFs and AUPR are considered to be more informative than AUC given the extent of class imbalance inherent in VS, but AUC is still the most widely employed parameter in the assessment of the screening power.[77]

When it comes to the ML-based SFs, only the scoring power or the screening power is evaluated in most of the related publications due to the fact that an ML-based SF is specifically trained for binding affinity prediction or VS, and therefore, the practicability and reliability of these SFs may not be fully guaranteed. For example, Gabel et al.[78] assessed the performance of two ML-based SFs (RF and SVM) that have been trained based on the simple protein–ligand element–element distance counts of the complexes in the PDBbind dataset. Although these two SFs displayed a significantly better scoring power than Surflex-Dock, they showed much poorer docking power and screening power, suggesting that a comprehensive assessment is quite necessary to an ML-based SF.

4 | WORKFLOW TO DEVELOP A MACHINE LEARNING-BASED SCORING FUNCTION

Like most other applications of ML, a training set and a test set are generally needed to develop an ML-based SF. However, how to construct an appropriate training/test set still remains in dispute. Some earlier methods, such as RF-Score,[79] purely utilized the core set of PDBbind as the test set and the remaining part of the refined set of PDBbind as the training set, but proteins from the same family could still be included in both the training and test sets. Kramer and Gedeck [80] doubted that the high structural overlap between the training and test sets might over-estimate the performance of RF-Score and instead they proposed an approach called leave-cluster-out cross-validation (LCOCV), where the whole dataset was clustered first and then split based on the clusters to avoid the influence of structural similarity or overlaps between the training and test sets.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

Surprisingly, RF-Score performed much worse than before, so they concluded that LCOCV might be more appropriate for SF development. Li and Yang[81] explored the impact of structural and sequence similarity on the prediction accuracy of ML-based SFs, and they found that the performance of RF-Score largely became worse with the decrease of the structural and sequence similarity between the training and test sets, but that of the empirical X-score was relatively stable. However, Ballester, the author of RF-Score, doubted that that LCOCV might be ultimately of little practical value, because these similarities could not be ignored when a SF was employed in a real scenario.[82] In addition, by conducting a similar study as Li et al.[83,84] did, they concluded that ML-based SFs could also learn from dissimilar training complexes, because RF-Score was able to outperform X-Score even when it was trained based on just 32% of the most dissimilar complexes. Thus, in our opinion, in order to develop a reliable SF, both the conventional random training/test splitting and LCOCV are needed, so that a comparison can be made to gain a deeper understanding of the impact of the SF itself and the composition of the dataset on the performance of a SF.

Once the dataset is determined, a set of features are generated to characterize the protein–ligand interactions for the complexes in the training and test sets. Traditionally, the features are individual energy terms used in classical SFs,[85–87] structural interaction fingerprints (SIFts),[88] etc., while in some recently developed DL-based SFs,[89–91] more basic structural information, such as atoms and bonds, has attracted more attention. Next, some conversion and filter processes such as feature selection (FS)[92] are used to remove redundant or irrelevant features, and then an ML algorithm is employed to train the SF based on the features for the training set and its actual performance is evaluated by the predictions to the test set. The model training process is along with the tuning of parameters, for which the training set is further split into two subsets, one for training and the other for validation.[93] In general, feature representation and ML are the two essential steps in the development of an ML-based SF, and the performance of a SF is highly dependent on the quality of these two steps.

5 | TRADITIONAL MACHINE LEARNING METHODS IN SCORING FUNCTIONS

Traditional ML methods, which are defined to be relative to DL in this study, have been widely employed in the development of SFs, whichever they are designed for binding affinity prediction or VS, or whichever they belong to generic SFs or target-specific ones. Some important applications of traditional ML methods in SFs are summarized in Table 2, and the following are the details, which are organized based on the types of ML methods.

5.1 | Random forest

RF is an ensemble learning method that introduces the bagging and subset selection strategies to multiple decision trees (DTs).[53] Each tree is trained by randomly sampled subsets instead of the original dataset, and then a consensus score is produced accordingly by integrating the outputs from multiple DTs. RF is a pioneer ML method used in SFs, and a variety of RF-based SFs, such as RF-Score,[79,88,94] B2BScore,[95] and SFCscore^{RF}[86] have been developed and outperformed a broad range of classical SFs. When it comes to RF-Score, there is no doubt that it has ushered in a new era in the context of SFs, despite the fact that a cloud of controversy has been thrown upon it.[78,80,81] Up to now, three versions of RF-Score have been developed. RF-Score-v1 intrinsically belongs to knowledge-based SFs, in which the number of occurrences of a certain protein–ligand atom type pair within a given distance range was simply counted and used as the features.[79] In total, nine common elemental atom types (C, N, O, S, P, F, Cl, Br, and I) were defined and finally 36 features were obtained after removing 45 ones whose values were equal to zero. Then, the PDBbind v2007 was used to train and test the model, and the best R_p for the test set could reach up to .776, which was much better than the classical SFs assessed before. RF-Score-v2 gained more improvement by tuning the parameters related to the representation of interatomic pairs, FS, model selection strategy, etc.[88] It was found that the two more complicated representations, SYBYL atom types and SIFts generated from CREDO,[124] did not outperform the simple element-based representation. Ultimately, by using the FS to remove sparse features and the Out-Of-Bag strategy to optimize the RF model, the best model could yield a R_p of .803. RF-Score-v3 not only inherited 36 features from v1 but also brought in six extra empirical energy terms computed by the Autodock Vina SF.[94] The final assessment on the PDBbind benchmark showed that this version could yield a considerable scoring power as v2 (R_p = .803). Besides, a recent study indicated that RF-Score-v3 outperformed X-Score even when 68% of the most similar proteins were removed from the training set, which further verified its superiority.[83,84]

B2BScore proposed by Liu et al. was trained based on two physicochemical properties, β contacts and B factors.[95] β contact was defined as a type of atomic contact between two atoms that have enough direct contact to form an important

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

TABLE 2 Summary of traditional machine learning methods used in scoring functions (SFs)

Authors	Year	SF[a]	Feature representation	Model	Applications[b]	Reference
Ballester and Mitchell	2010	RF-Score	Counts of elemental atom pairs	RF	Scoring	79
Ballester et al.	2014	RF-Score-v2	Counts of elemental atom pairs	RF	Scoring	88
Li et al.	2015	RF-Score-v3	Terms from RF-Score and Vina	RF	Scoring	94
Li et al.	2014	/[c]	Terms from Cyscore, Vina and RF-Score	RF	Scoring	87
Liu et al.	2013	B2BScore	β contacts and B factor	RF	Scoring	95
Zilian and Sotriffer	2013	SFCscore^{RF}	Terms from SFCscore	RF	Scoring	86
Ashtawy and Mahapatra	2012	/	Different combinations of the terms from X-Score, AffiScore and RF-Score	RF, BRT, SVM, kNN, MARS or MLR	Ranking	96
Ashtawy and Mahapatra	2015	/	Different combinations of the terms from X-Score, AffiScore and RF-Score	RF, BRT, SVM, kNN, MARS or MLR	Scoring	85
Ashtawy and Mahapatra	2015	/	Different combination of the terms from X-Score, AffiScore, GOLD and RF-Score	RF, BRT, SVM, kNN, MARS or MLR	Pose prediction	97
Khamis and Gomaa	2015	/	All terms from RF-Score, BALL, X-Score and SLIDE	RF, BRT, kNN, SVM and etc.	Scoring, ranking, pose prediction and VS	98
Wojcikowski et al.	2017	RF-Score-VS	Terms from three versions of RF-Score	RF	VS	99
Nguyen et al.	2017	RI-Score	Element-specific rigidity index	RF	Scoring	100
Wang and Zhang	2017	$\Delta_{vina}RF_{20}$	10 terms from Vina and 10 terms related to bSASA	RF	Scoring, ranking, pose prediction and VS	101
Yasuo and Sekijima	2019	SIEVE-Score	Per-residue interaction energy from Glide	RF	VS	102
Li et al.	2011	SVR-KB	Knowledge-based pairwise potentials	SVM	Scoring	103
		SVR-EP	Four features extracted from 14 empirical descriptors by FS	SVM	Scoring	
		SVR-KBD	Knowledge-based pairwise potentials	SVM	Target-specific VS	
Li et al.	2011	SVM-SP	Knowledge-based pairwise potentials	SVM	Target-specific VS	104
Xu and Meroueh	2016	SVMGen	Knowledge-based pairwise potentials	SVM	VS (kinase)	105
Koppisetty et al.	2013	/	Different combinations of the descriptors generated from Liaison, Embrace, prime-MMGBSA, Glide, Qikprop and Ligparse modules	SVM	Scoring	106
Li et al.	2013	ID-Score	50 protein–ligand interactions-related descriptors	SVM	Scoring	107
Das et al.	2010	PESD-SVM	Molecular shapes and property distributions	SVM	Target-specific VS	108
Sato et al.	2010	Pharm-IF	Pharmacophore-based interaction fingerprint	SVM, NBC, RF or ANN	Target-specific VS	109
Ding et al.	2013	MIEC-SVM	Terms from Glide and SASA	SVM	Target-specific VS	110
Yan et al.	2017	PLEIC-SVM	Protein−ligand empirical interaction components	SVM	Target-specific VS	111
Nogueira and Koch	2019	PADIF-SVM (NN)	Interaction fingerprint derived from GOLD	SVM, ANN	Docking-based target prediction	112

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

TABLE 2 (Continued)

Authors	Year	SF[a]	Feature representation	Model	Applications[b]	Reference
Durrant and McCammon	2010	NNScore	Knowledge-based pairwise potentials	ANN	VS	113
Durrant and McCammon	2010	NNScore 2.0	12 features from BINANA and 6 terms from Vina	ANN	VS	114
Ouyang et al.	2011	CScore	Elemental atom pairs transformed by two fuzzy membership functions	Modified CMAC network	Scoring	115
Arciniega and Lange	2014	DDFA	*DockScore*, *DockLE*, *DockSimi*, *DockPoses* and *DockRmsd*	ANN	VS	116
Ashtawy and Mahapatra	2016	BgN-Score BsN-Score	All terms from X-score, AffiScore, GOLD and RF-score	BgN BsN	Scoring	117
Ashtawy and Mahapatra	2017	BT-Score BT-Dock BT-Screen	2,700 descriptors from DDB	GBDT	Scoring Pose prediction VS	118
Wang et al.	2017	FFT-BP	Six types of microscopic features	GBDT	Scoring	119
Cang and Wei	2017	T-Bind	Topological fingerprints	GBDT	Scoring	120
Cang et al.	2018	TopBP TopVS	Topological fingerprints	A consensus model of GBDT and CNN	Scoring VS	121
Nguyen et al.	2019	EIC-Score	Element interactive curvatures (EICs)	GBDT	Scoring	122
Li et al.	2019	XGB-Score	Terms from RF-Score and Vina	GBDT	Scoring	84
Pires and Ascher	2016	CSM-lig	Cutoff scanning matrix (CSM)	GP	Scoring	123

[a] In most cases, only the model with the best performance is listed in the table.

[b] As the training and test sets for each method are not always the same, the concrete value of each performance is not listed in the table.

[c] It means this method is not named or an assessment has been conducted in this study.

interaction. B factor, which is also known as the temperature factor, is an important property for the measurement of the mobility and flexibility of atoms in proteins. Tested on the PDBbind v2009, B2BScore showed superior performance (R_p = .749) than the other tested SFs and it also achieved a significant LCOCV improvement to RF-Score-v1 across 26 protein clusters with R_p increasing from .418 to .518.

Nguyen et al. believed that ligand binding should reduce protein flexibility and strengthen protein rigidity, so they proposed a SF named RI-Score based on the rigidity index.[100] This method considered the change of protein rigidity upon ligand binding, and the element-specific rigidity indices were calculated directly from atom–atom distances and used as the features in model development. This SF was tested on the PDBbind v2007, v2013 and v2016 core set, and the R_ps are .803, .762 and .815, respectively.

There are also some RF-based SFs optimized from classical empirical SFs by just using the RF to replace the original linear fitting method.[78,85,86,94,96–98] For example, SFCscore^{RF} was developed based on the SFCscore[125] descriptors,[86] and it showed significantly better scoring power than SFCscore (R_p = .779 vs. .644), but the LCOCV results indicated that its performance was also highly target-dependent. However, in another assessment on the CSAR 2012 exercise datasets for three targets (CHK1, ERK2, and LpxC), the predictions of SFCscore^{RF} for CHK1 and ERK2 were below the expectations, suggesting that the applicability domain of SFCscore^{RF} needs more extensive verification.[86]

Besides SFCscore, other classical SFs, such as Cyscore, Vina, AffiScore and X-Score, were also employed to develop RF-based SFs, and a remarkable improvement could be observed in all situations.[85,87] In addition, different combinations of different descriptors of classical SFs were also tried to construct ML-based SFs. Ashtawy and Mahapatra carried out a comparative assessment of the prediction accuracies of classical and ML-based SFs for protein–ligand binding affinity prediction.[85] A number of ML-based SFs were developed by six ML techniques based on a variety of energetic, physicochemical and geometrical features computed by X-Score, AffiScore and RF-Score. The results showed that the best RF-based SF (R_p = .806) outperformed the best classical SF (R_p = .644) for the PDBbind v2007 core set. Another assessment reported by the same group indicated that ML-based methods could achieve a better ranking power than classical SFs and the ranking rate for the best RF-

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

based on SF employing the X-Score and AffiScore features could reach up to 62.5%.[96] Furthermore, they also employed the similar strategy to evaluate the docking power, but unfortunately most ML-based SFs did not perform as well as classical ones just based on the top-ranked poses,[97] which was consistent with the results reported by Gabel et al.[78]

Most RF-based SFs are designed for binding affinity prediction, which may partially owe to the fact that they are possibly not so suitable for VS at all. Actually, some SFs specially developed for VS have also been reported. For example, Wojcikowski et al. proposed a RF-Score-based SF named RF-Score-VS for VS.[99] The binding poses for the ligands in the DUD-E dataset were generated by AutoDock Vina, Dock6.6 and Dock3.6, respectively, and then three dataset partitioning strategies, including per-target (similar with LCOCV), horizontal split (both training and test sets contained data from all targets) and vertical split (some targets were used as the training set and the others as the test set), were used to construct the training and test sets. Then, the RF-Score-VS SFs were trained based on the descriptors used in the three versions of RF-Score. The final results indicated that the RF-Score-VS SFs have much better performance than the tested classical SFs for the per-target and horizontal dataset partitioning strategies, but it does not show any improvement for the vertical split dataset partitioning strategy, suggesting that RF-Score-VS might not have good capability to discover ligands for completely novel targets. Another application of RF in VS was the exploitation of Similarity of Interaction Energy VEctor Score (SIEVE-Score),[102] which was inspired by the interaction fingerprint approaches and was trained based on the per-residue van der Waals, Coulomb, and hydrogen bonding energy terms extracted from the Glide SP docking results. Using the DUD-E as the training set and the DEKOIS 2.0 as an independent test set, SIEVE-Score not only showed better performance than Glide, Dock and AutoDock Vina, but also outperformed the above-mentioned RF-Score-VS. Besides, this method could also show better screening power than some interaction fingerprints, including SPLIF, PLIF and ligand similarity based on the functional connectivity fingerprints up to the second closest neighbor (FCFP4).

Wang and Zhang[101] proposed a RF-based SF called $\Delta_{\text{vina}}RF_{20}$, and the most pleasant thing was that its performance was comprehensively assessed by the CASF-2007 and CASF-2013 benchmarks. Different from most SFs that were developed by fitting the final binding scores to the experimental data with RF, the fitting strategy of this method was to employ RF to parameterize corrections to the AutoDock Vina scores, which meant that $\Delta_{\text{vina}}RF_{20}$ was calculated as the sum of the AutoDock Vina score and the correction term trained by RF. In addition, unlike using only PDBbind as the training set in most SFs, $\Delta_{\text{vina}}RF_{20}$ was trained based on three datasets, including the PDBbind-v2014 refined set, the native poses in the CSAR decoy dataset and the weak binding structures in the PDBbind-v2014 general set. The $\Delta_{\text{vina}}RF_{20}$ SF, which was developed based on 10 AutoDock Vina terms determined by FS and the other 10 terms related to buried solvent-accessible surface area, achieved better performance than most assessed classical SFs in both the CASF-2007 and CASF-2013 benchmarks, suggesting that optimizing the correction term with ML algorithms might be a promising way to efficiently improve the performance and robustness of SFs.

5.2 | Support vector machines

SVMs developed by Vapnik and cowokers[54,55] are a set of supervised learning methods that are capable of handling high-dimensional variables for small datasets.[54,55] SVM is originally designed for classification, but its derivative support vector regression (SVR) can be used for regression. SVMs solve the classification problems by using nonlinear kernel functions to map data into high-dimensional space by finding an optimally separating hyperplane, whereas the regression is achieved by searching a hyperplane with the optimized sum of the distances from the data points to the hyperplane.[126] Four kernel functions, including linear, polynomial, sigmoid, and radial basis function (RBF) are usually used in conventional SVM modeling, of which the RBF kernel was considered to be the most robust and thereby the most widely adopted kernel. SVMs were also commonly used in the development of SFs, especially the target-specific SFs for VS.[103–111,127,128]

Koppisetty et al.[106] built the SVR-SFs for binding affinity prediction by utilizing several protein–ligand interaction or ligand-based descriptors generated from the Liaison, Embrace, prime-MMGBSA, Glide, Qikprop and Ligparse modules implemented in Schrödinger. The model developed based on all these descriptors yielded the best performance, and the R_p values for the PDBbind v2002 dataset, v2007 core set and v2009 refined set were .612, .600 and .630, respectively. Besides binding affinity prediction, this model could also estimate the enthalpy (ΔH) and entropy (ΔS) components with acceptable reliability. Similarly, Li et al.[107] developed a SVR-based SF-named ID-Score based on 50 descriptors related to protein–ligand interactions. The evaluation results showed that ID-Score achieved a higher scoring power ($R_p = .753$) than the other tested commonly used SFs on the PDBbind v2007 and also exhibited a better ability in differentiating structurally similar ligands than the other seven SFs.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

Meroueh's group proposed five SVM-based SFs, namely SVM-SP, SVR-KB, SVR-EP, SVR-KBD, and SVMGen.[103–105] Both of SVR-KB and SVR-EP were trained for binding affinity prediction, and the difference was that the former was established based on the knowledge-based pairwise potentials using the SYBYL atom types extracted from protein–ligand complexes and the latter was established based on four physicochemical properties extracted from 14 empirical descriptors by FS. SVR-KB and SVR-EP were trained by various training sets, and tested by the CSAR-SET2 or CSAR-SET1 datasets, and SVR-KB achieved the best scoring power ($R_\mathrm{p}^2 = .67$) for CSAR-SET2. Both of SVR-SP and SVR-KBD were designed for target-specific VS and established based on distance-dependent pairwise potentials. The assessment on the DUD dataset indicated that SVR-KBD did not exceed the screening power of SVM-SP, but still exhibited higher enrichment than other tested SFs. However, SVMGen, which was established with the same strategy as SVM-SP except that the negative set was composed of a collection of randomly selected compounds docked to a diverse set of protein structures, was reported as a generic SF specially designed for kinases. Unfortunately, SVMGen did not perform always as well as GlideScore, but its superior performance on homology models where more less accurate binding poses might be generated suggested that it might be useful in the screening campaigns towards targets without crystal structures.

In addition to the SFs mentioned above, SVMs were also used in other target-specific methods that were served as post-docking filters, such as PESD-SVM,[108] Pharm-IF,[109] MIEC-SVM,[110] PLEIC-SVM[111] and PADIF-SVM (NN).[112] One common characteristic of these methods is that they all adopt a per-residue decomposition to generate some interaction fingerprint-like features to represent protein–ligand interactions. PESD-SVM refers to the SVM model where the molecular shapes and property distributions on protein and ligand surfaces are featured. Pharm-IF is a method based on pharmacophore-based interaction fingerprints while PLEIC-SVM is established based on protein–ligand empirical interaction components. MIEC-SVM and PADIF-SVM (NN), however, were mainly derived based on the energy terms of Glide and GOLD, respectively. To date, several small-molecule inhibitors have been discovered by the target-specific scoring methods based on SVM, which verified their capability in drug discovery.[104,129]

5.3 | Artificial neural networks

Artificial neural networks (ANNs),[130] which were originally designed to model brain structure and functioning, are regarded as a class of general and flexible ML methods. The network is generally made up of several simple neurons, which are arranged in a particular topology and connected to each other. Then, neurons are organized into layers, which can be divided into input layers, hidden layers and output layers based on their locations. Among all the traditional ANNs, back propagation neural network (BPNN) proposed by Rumelhart et al.,[56] which is applied to multiplayer perceptrons, is considered to be the most popular and well-studied training algorithm. It is a gradient-descendent method that minimizes the mean square error of the difference between the network outputs and the data in the training set. Although traditional shallow ANNs are always related with poor generalization performance for higher dimensional data, there is no doubt that they also contribute a lot to the development of ML-based SFs.[117]

NNScore proposed by Durrant and McCammon was the first NN-based SF and designed as a binary classifier to distinguish good and poor binders.[113] Five types of knowledge-based pairwise potentials were defined to characterize the close contacts, semi-close contacts, electrostatic-interaction energy, number of ligand atom types and number of ligand rotatable bonds, and finally 194 features were obtained in total. The final networks were constructed by using feed-forward networks with 194 inputs and two outputs. NNScore 2.0 gained further improvement over the first version by incorporating more binding features, including 12 binding feature generated by BINANA (an expansion of the features used in NNScore 1.0)[131] and other six energy terms implemented in AutoDock Vina 1.1.2, and it could support the quantitative estimate of pK_d rather than pure binary classification.[114] As for the networks, NNScore 2.0 used a single hidden layer of 10 neurons to replace the original five neurons, and the number of the output layer was reduced to 1 which just corresponded to the predicted pK_d. Further assessment on the DUD dataset showed that NNScore 2.0 did not perform as well as NNScore 1.0 on average, but outperformed AutoDock, AutoDock Vina and Glide HTVS.[132] In addition, NNScore was also involved in the discovery of inhibitors towards several important targets, which further testified its practicality.[133–137]

Ouyang et al.[115] developed a method named CScore that was trained by using a modified Cerebellar Model Articulation Controller learning architecture for binding affinity prediction. Similar with RF-Score, this method also relied on simple element–element atom pairs but used fuzzy membership functions rather than directly counting the occurrence to represent each feature. This method could achieve R_ps of .801 and .767 for the PDBbind v2007 and v2009 core sets, respectively, and the LCOCV results also exhibited its superiority.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

Arciniega and Lange[116] proposed an ANN-based rescoring tool referred to as docking data feature analysis (DDFA) for VS. Five types of features, namely *DockScore*, *DockLE*, *DockSimi*, *DockPoses* and *DockRmsd*, were obtained from the docking results of AutoDock, AutoDock Vina or Rosetta Ligand. Then, a feed-forward ANN, which consisted of 13, 8, and 1 neurons for the input, hidden and output layers, respectively, was constructed, and a leave-one-out (LOO) cross-validation towards DUD was conducted to test each model. The assessment results showed that DDFA with all the features provided by the three docking programs yielded the best screening power, which was similar to the performance given by the best available method.

Inspired by the outstanding performance of ensemble learning methods such as RF and boosted regression trees in a previous assessment study,[85] Ashtawy and Mahapatra[117] also proposed two ensemble NN-based SFs based on bagging (BgN-Score) and boosting (BsN-Score). To make a better comparison, the same combinations of the features used in the previous assessment were used in this study. The evaluation results on the PDBbind v2007 core set demonstrated that BSN-Score and BgN-Score based on all the X-Score, AffiScore, GOLD and RF-Score features (R_p = .816 and .804, respectively) outperformed the previously proposed RF-based SF (R_p = .801) and the best of the tested conventional SFs (R_p = .644).

5.4 | Gradient boosting decision tree

Gradient boosting decision tree (GBDT) is also an ensemble learning method that uses DTs as the base learners.[138] It builds the model by combining weak base learners into a single strong learner in an iterative way, and then generalizes them by allowing an optimization of an arbitrary differentiable loss function. Extreme gradient boosting (XGBoost) proposed by Chen and Guestrin,[139] an efficient and scalable implementation of the gradient boosting framework, has been regarded as a new generation of ensemble learning algorithm, and has become the winners for several ML competitions in recent years.[140–142] Besides, GBDT has been widely employed in the field of drug discovery and recently also involved in the development of novel SFs.[118–122]

XGB-Score, which originated from RF-Score, was developed by XGBoost instead of RF based on the descriptors from RF-Score-v3.[84] To overcome the stochasticity of the training processes of ML, 10 instances were generated and the final performance was assessed by averaging their predictions. As a result, besides the classical SFs such as X-Score and Cyscore, XGB-Score even outperformed RF-Score-v3 in most cases (average R_p = .806 vs. .800 on the PDBbind v2007), which further verified the superior learning capability of XGBoost.

Ashtawy and Mahapatra[118] proposed three XGBoost-based SFs, namely BT-Score, BT-Dock and BT-Screen, which were designed for binding affinity prediction, binding poses prediction and VS, respectively. These SFs were developed based on ~2,700 multiperspective descriptors generated by Descriptor Data Bank.[143] The PDBbind v2014 was used as the major training set, and a number of computer-generated ligand conformations and inactive protein–ligand complexes were added into the training sets for BT-Dock and BT-Screen, respectively. The testing results on CASF-2013 indicated that BT-Score, BT-Dock and BT-Screen could yield the higher scoring power (R_p = .825 vs. .627), docking power (S_2 = 96.87% vs. 82.05%) and screening power ($EF_{1\%}$ = 33.90 vs. 19.54) than other conventional SFs, despite the fact that BT-Score did not perform well in terms of the docking power and screening power.

Wang et al.[119] developed a feature functional theory-binding predictor (FFT-BP) for protein–ligand binding affinity prediction based on six types of microscopic features, including reaction field features, electrostatic binding features, atomic coulombic interaction, atomic van der Waals interaction, atomic solvent-excluded surface area and molecular volume. The final results of R_p on the PDBbind v2007, v2015 and v2016 core sets could reach .800, .780 and .747, respectively. Later, the same group also developed three other GBDT-based SFs, namely T-Bind,[120] TopBP-ML,[121] and EIC-Score.[122] Unlike most methods that used geometry-based features to represent protein–ligand interactions, T-Bind revolutionarily introduced the element specific persistent homology (ESPH) method to extract features. ESPH could embed multiscale geometric information into topological invariants and help to decipher the entangling codes of protein–ligand interaction. The barcodes generated from the ESPH calculations were called the topological fingerprints, which deemed to be a suitable feature representation that could be used to develop the following ML models. The final method was assessed on the PDBbind v2007, v2013 and v2015 core sets, and the R_ps were .818, .767, and .775, respectively. Similarly, TopBP-ML also employed an algebraic topology-based approach, but besides ESPH, it further brought in multilevel persistent homology and electrostatic persistence for the description of protein–ligand interactions. Its average scoring power on the PDBbind v2007, v2013, v2015, and v2016 core sets was slightly better than that of the CNN-based TopBP-DL (R_p = .817 vs. .809), but worse than that of TopBP (taking the average of TopBP-ML and TopBP-DL), which achieved R_ps of .827, .808, .812, and .861 on the four versions of PDBbind, respectively (the average R_p = .827). The same feature representation strategy was also used to develop the SF for VS, and

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

similarly a consensus method, TopVS, could achieve the best screening power on the DUD benchmark, with AUC and $EF_{2\%}$ equaling to 0.84 and 9.5, respectively. EIC-Score, however, was developed based on a molecule representation method called the differential geometry-based geometric learning (DG-GL) hypothesis, whose assumption was that the intrinsic physics lay on a family of low-dimensional manifolds embedded in a high-dimensional data space. In brief, an element-level coarse-grained representation (element interactive densities) was firstly extracted from protein–ligand complexes, followed by the generation of a few element interactive manifolds. Then, based on these differentiable manifolds, the associated element interactive curvatures (EICs) were generated accordingly and featured for the final model construction. The assessment results on the PDBbind v2007, v2013, and v2016 core sets showed that the best R_ps were .817, .774, and .828, respectively.

5.5 | Gaussian process

Gaussian process (GP)[144] is defined as a natural generalization of multivariate Gaussian random variables to a Gaussian distribution over a specific family of functions (such as covariance function or kernel).[14] It represents a flexible Bayesian nonparametric approach to solve nonlinear optimization problems that is able to gracefully deal with uncertainty, uneven sampling and a diverse range of dynamic behaviors. Pires et al. used GP to train and develop a web server named CSM-lig for assessing and comparing protein–ligand affinities.[123] One of the highlights of this method was that a class of graph-based signatures, called cutoff scanning matrix (CSM) based on the atomic distance patterns surrounding a bound ligand, was employed to represent the 3D environments for proteins and ligands. The assessment on the PDBbind v2007 core set showed that CSM-lig could achieve an acceptable scoring power ($R_p = .751$), which was much better than most classical SFs.

6 | DEEP LEARNING IN SCORING FUNCTIONS

The concept of DL is originated from ANNs, where feedforward NNs combined with larger numbers of hidden layers are considered as deep NN.[145] Compared with traditional ANNs, DL can afford many more neurons in each layer due to the appearance of more powerful CPU and GPU hardware. Some algorithmic improvements significantly enhance the practicality and reliability of DL, such as the application of the dropout method[146] to address the overfitting problem and the use of the rectified linear unit[147] to avoid vanishing gradients. Popular implements of DL include fully connected deep neural network (DNN),[148] convolutional neural network (CNN),[149] recurrent neural network,[150] and autoencoder,[151] among which CNN emerges as the most widely used DL method to build SFs. Table 3 summarizes some recent applications of DL in SFs.

6.1 | CNN in structure-based VS

CNN, which is commonly applied to image recognition, consists of an input layer and an output layer, as well as multiple hidden layers. The hidden layers of a typical CNN can be classified into convolutional layers, pooling layers and fully connected layers, in which the convolutional layers are used to extract features from the input, while the pooling layers subsample or downsample feature maps typically with average, sum, or max pooling. Fully connected layers, however, connect every neuron in one layer to every neuron in another layer just like in a traditional ANN.

AtomNet proposed by Wallach et al. was considered as the first application of CNN for binding affinity prediction, and also the first DL architecture that involved the structural information of targets to achieve its prediction.[152] This method received vectorized versions of 1 Å 3D grids placed over the binding site of the target as the input, where multiple poses of a ligand were sampled and each grid cell held a value to represent the presence of some basic structural features. Then, four 3D convolutional layers and two fully connected layers were constructed, and finally a logistic regression cost layer was utilized to output two activity classes. The method was tested on multiple VS-related datasets, and the average AUC on DUD-E achieved the value of 0.895, which was much higher than that of Smina (AUC = 0.696).

Besides AtomNet, there are also other novel CNN-based SFs designed for VS. For example, Pereira et al.[153] developed an approach named DeepVS, which automatically learned to extract relevant features from basic structural data (such as atom types, atomic partial charges, distances of atom pairs, and amino acid types) to improve the results of docking calculations (including AutoDock Vina and Dock 6.6). An innovation of DeepVS was that it introduced the concept of atom and amino acid embeddings, which could easily incorporate the protein–ligand complex raw data into a DNN. The network totally contained three hidden layers, where the first two aimed to extract features for the representation of atom context and protein–ligand interactions, respectively, while the last one along with the output layer was employed to compute a score for each complex. The evaluation results on DUD showed that DeepVS outperformed the pure docking programs, and the model based

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

TABLE 3 Summary of deep learning methods used in scoring functions (SFs)

Authors	Year	SF[a]	Feature representation	Model	Applications[b]	Reference
Wallach et al.	2015	AtomNet	Some basic structural features	CNN	VS	152
Pereira et al.	2016	DeepVS	Atom types, charges, distances and amino acid types	CNN	VS	153
Ragoza et al.	2017	/	Atom type information with the form of a density distribution	CNN	Pose prediction, VS	90
Ragoza et al.	2017	/	Atom type information with the form of a density distribution	CNN	Pose optimization	154
Imrie et al.	2018	DenseFS	Atom type information with a Gaussian representation	DenseNet	VS	91
Gonczarek et al.	2017	/	Basic features related with atom and atom connections	CNN	VS	155
Gomes et al.	2017	/	Atomic coordinates and atom types	CNN	Scoring	89
Feinberg et al.	2018	PotentialNet	Atom types, atomic distances and bonds	GCN	Scoring	156
Cang and Wei	2017	TNet-BP	Topological fingerprints	CNN	Scoring	157
Jimenez et al.	2018	K_{DEEP}	A particular property channel	CNN	Scoring	158
Stepniewska-Dziubinska et al.	2018	Pafnucy	Atomic coordinates and some features associated with basic atoms, bonds or partial charges	CNN	Scoring	159
Ashtawy and Mahapatra	2017	MT-Net	2,700 descriptors from DDB	MTL	Scoring, pose prediction and VS	118

[a] In most cases, only the model with the best performance is listed in the table.

[b] As the training and test sets for each method are not always the same, the concrete value of each performance is not listed in the table.

on the output from AutoDock Vina could yield the best prediction (AUC = 0.81). Ragoza et al.[90] also discretized the protein–ligand structure into a 3D grid to handle the features, but they represented atom type information as a density distribution around the atom center, where a continuous piecewise combination of a Gaussian function and a quadratic function was utilized to describe this representation. Five 3D convolutional layers alternating with max pooling layers constituted the model architecture, and several relevant model parameters were explored to achieve the best predictions. The method was trained with DUD-E and tested on the other two independent test sets, and a cluster cross-validation on DUD-E could yield an average AUC of 0.868, but the results on the other two datasets did not perform as well. Following the work reported by Ragoza et al., Imrie et al. [91] also proposed a CNN-based method called DenseFS, which employed the same input format but differed in four key ways from the previous study, including using densely connected CNN (DenseNet)[160] to replace the original shallow CNN, averaging the top *n*-ranked poses from AutoDock Vina rather than the top-ranked ones, adopting the strategy of transfer learning to construct protein family-specific models, and employing an ensemble of models trained with different random seeds to improve performance. The AUC and AUPR for the cluster cross-validation on DUD-E were 0.917 and 0.443, respectively, which outperformed AutoDock Vina (0.703 and 0.093) and a baseline CNN method based on the study reported by Ragoza et al. (0.862 and 0.263). In addition, the validation results on the external ChEMBL and MUV test sets also highlighted the superiority of this method. Unlike the above approaches that generated a 3D grid in advance, Gonczarek et al.[155] directly adopted a CNN-improved extended connectivity fingerprint for the construction of their VS-aimed SF,[161] where an innovative approach named atom convolution was conducted to convert simple atom features including types and partial charges into a fingerprint to represent each complex. They believed that traditional DUD-E might be strongly biased due to the similarity of artificial decoys for different targets, so the model was trained on PDBbind and tested on the DUD-E and MUV datasets. The assessment results on DUD-E and MUV suggest that the performance (AUCs = 0.704 and 0.575) is better than those of AutoDock Vina (0.633 and 0.503) and GCN (0.567 and 0.474), another similar method based on graph convolutional networks.

6.2 | CNN in binding affinity predictions

Gomes et al.[89] first proposed the atomic CNN to scoring, and the main steps lay on the use of two new convolutional operations, namely atom type convolution and radial pooling, which intended to extract features from the input representation by

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

making use of a neighbor-listed distance matrix and down-sampling the output of the atom type convolution. The PDBbind v2015 core set or refined set was split into the training and test sets with the method of random, stratified, scaffold or temporal, and all the splitting followed a ratio of 80/20. This method did not perform as well as the other tested methods, but the introduction of atomic convolutions to SFs was still of great significance to the improvement of the scoring power. Before long, the same group proposed a graph CNN (GCN)-based SF named PotentialNet.[156] This approach consisted of three major steps to achieve feature learning, including covalent-only propagation, dual noncovalent and covalent propagation, and ligand-based graph gather, and the input features were only basic information about atoms, bonds, and distances. The assessment on the PDBbind v2007 with the typical *refined train and core test* mode illustrated that PotentialNet (R_p = .822) outperformed RF-Score (R_p = .783) and X-Score (R_p = .643). But when the other two sequence and structure-based cross-validation strategies were used, the R_p values of PotentialNet became .700 and .823, respectively, and the former was even lower than that of RF-Score (R_p = .732), which further highlighted the importance of dataset partitioning. Driven by the excellent performance of the ESPH-based T-Bind, Cang and Wei further improved the model by using CNN to replace the original GBDT method.[157] By integrating the advantages of ESPH and CNN, a multichannel topological neural network named TopologyNet (TNet) was proposed, and accordingly a TNet-binding predictor (TNet-BP) was also developed for binding affinity prediction. The barcodes were firstly transformed to a one-dimensional (1D) image-like representation with multiple channels, and then a network composed of a few 1D convolution layers and several fully connected layers was used to extract higher level features from topological images and to perform regression with the learned features. The assessment could yield the R_ps of .826 and .81 for the PDBbind v2007 and v2016 core sets, respectively. Other recently developed CNN-based SFs included K_{DEEP} proposed by Jimenez et al.[158] and Pafnucy proposed by Stepniewska-Dziubinska et al.,[159] and both of them were established by using a set of descriptors based on 3D grids. In K_{DEEP}, a particular property channel (including hydrophobic, hydrogen bond donor or acceptor, aromatic, positive or negative ionizable, metallic and total excluded volume) was used to characterize each atom to each grid for both ligands and proteins. This method could yield a R_p of .82 to the PDBbind v2016 core set, but it only got an average R_p of .59 to other four individual CSAR datasets, which was not as good as several other tested SFs, such as RF-Score-v3 (R_p = .7) and X-score (R_p = .64). As for Pafnucy, a four-dimensional tensor in which each point was defined by the three vectors of Cartesian coordinates and a vector of features associated with basic atoms, bonds or partial charges were used as the representation of the input, and the network consisted of three 3D convolutional layers followed by max pooling layers and three fully connected layers.[159] This model was tested with the protein–ligand complexes in the PDBbind v2016 core set, PDBbind v2013 core set and Astex diverse set, yielding the R_p values of .78, .70, and .57, respectively, which were better than those of X-Score.

6.3 | CNN in binding pose predictions

CNN was also employed to distinguish correct and incorrect binding poses. For example, besides the exploration of VS, Ragoza et al.[90] also tried their CNN model in pose prediction. The ligands extracted from the CSAR dataset were redocked with AutoDock Vina to generate enough poses as the training set, where the poses with RMSD less than 2 Å from the crystal pose were labeled as the correct poses while those with RMSD larger than 4 Å were regarded as the incorrect. In terms of AUC, their method performed much better than AutoDock Vina either based on cross-validation (0.815 vs. 0.645) or external test on the dataset with the poses generated from the PDBbind core set (0.792 vs. 0.682). But as to the success rates among the top *n* poses, the CNN-based method performed slightly worse, and especially when the top one pose was chosen to represent each ligand. Another study conducted by Ragoza et al.[154] further extended the application to pose optimization, and they attempted to use the gradient of their CNN-based SF to generate optimal binding poses. An iteratively trained CNN that included the poses optimized by the first CNN in its training set was designed to optimize the randomly initialized poses, and it performed better than simple CNN-based SFs and AutoDock Vina, though the performance was still far from expectation.

6.4 | Transfer and multitask learning

Transfer and multitask learning (TL and MTL) are two technologies that can make full use of the commonality of multiple tasks to build a more robust model.[162,163] Combined with DL, they display a promising prospect in the development of novel ML-based SFs. TL can be simply described as the ability of a model to recognize the knowledge learned from previous tasks and then apply it for the solution of new problems. Recently, Imrie et al.[91] developed an ensemble of protein family-specific models based on the DUD-E and ChEMBL benchmarks by coupling the densely connected CNN (DenseNet) with a TL approach. The use of TL largely reduced the training cost, and the obtained family-specific models could achieve a significant

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

improvement compared to a single universal model. MTL, however, refers to the approaches where several tasks are learned simultaneously within a single model. For example, Ashtawy and Mahapatra[118] proposed a multitask deep neural network (MT-Net) that was built from three individual tasks supported by BT-Score, BT-Dock, and BT-Screen to simultaneously predict binding affinities, poses, and activity levels. This network consisted of three shared hidden layers for each task to learn common features, a total of six task-specific hidden layers for high-level feature representation, and three task-specific output layers. The assessment results indicated that although MT-Net did not perform as well as the original BT-Score, BT-Dock, or BT-Screen, it surely showed higher performance than conventional SFs, and no poorer than single-task networks (Box 1).

BOX 1 Some ml-based sfs-related tools or websites

RF-Score-v2[88]: https://bitbucket.org/aschreyer/rfscore
idock[164]: http://istar.cse.cuhk.edu.hk/idock/
NNscore1.0/2.0[113,114]: http://rocce-vm0.ucsd.edu/data/sw/hosted/nnscore//
XGB-Score[84]: https://github.com/HongjianLi/MLSF
CSM-Lig[123]: http://structure.bioc.cam.ac.uk/csmlig/
$\Delta_{\text{vina}}\text{RF}_{20}$[101]: http://www.nyu.edu/projects/yzhang/DeltaVina/
RI-Score[100]: https://weilab.math.msu.edu/RI-Score/
TopBP[120]: https://weilab.math.msu.edu/TML/TML-BP/
TNet-BP[157]: https://weilab.math.msu.edu/TDL/TDL-BP/
EIC-Score[122]: https://weilab.math.msu.edu/DG-GL/
Pafnucy[159]: http://gitlab.com/cheminfIBB/pafnucy
K_{DEEP}[158]: https://playmolecule.org/
Descriptor Data Bank (DDB)[143]: http://www.descriptordb.com/
Open Drug Discovery Toolkit (ODDT)[165]: https://github.com/oddt/oddt/
DeepChem/MoleculeNet[77]: http://deepchem.io.s3-website-us-west-1.amazonaws.com/

7 | CONCLUDING REMARKS AND FUTURE PERSPECTIVES

Introducing ML methods to SFs has emerged as a promising trend in the context of drug design, and the application of more data-hungry DL methods further brings vigor and vitality into the development of ML-based SFs. In this study, we have provided a comprehensive review of ML-based SFs. Firstly, we have made a brief description of the classification of SFs, several important benchmarks to construct or assess the SFs, and the workflow to build an ML-based SF. Then, all the ML approaches have been roughly classified into traditional ML and DL methods. In terms of the former, we have mainly focused on five commonly adopted approaches including RF, SVM, ANN, GBDT and GP, whereas regarding the latter, CNN has been paid more attention.

Table 4 makes a comparison of the scoring power of different SFs assessed on the PDBbind core set, and Table 5 lists the screening power of some approaches tested on the DUD or DUD-E dataset. Compared with classical SFs, ML-based methods have always been found to achieve a remarkable improvement, whether towards simple binding affinity prediction or VS. Relatively speaking, ensemble methods such as RF and GBDT performed better than other traditional ML approaches, while SVM was more preferred to be used for target-specific VS. As for DL, besides the popularity of simple CNN, some derivatives such as DenseNets that introduced dense connections in networks and GCNs[166] that directly learned features from undirected graphs, could also efficiently exert their functions. An interesting finding was that DL-based methods did not always outperform traditional ML-based ones. Exactly, the performance of a SF may depend more on the ensemble effect of all key steps in the construction of a SF rather than simply the employed ML method. In addition to the change of model construction, some improvements in feature representation also contribute a lot to the development of SFs (Figure 2). SFs in the early stages tended to feature some physical interaction terms such as the van der Waals and electrostatic interactions that could be obtained directly from existing classical SFs and some basic physiochemical or structural descriptors. However, due to the vigorous feature extraction ability of the convolution layers, recently proposed CNN-based approaches just needed

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

TABLE 4 Scoring power (R_p) of different machine learning-based scoring functions (SFs) on the PDBbind core set

SF[a]	v2007[b]	v2009	v2013	v2016	Reference
TopBP	.827		.808	.861	121
TNet-BP	.826			.810	120
PotentialNet	.822				156
TopBP-ML	.818		.804	.848	121
EIC-Score	.817		.774	.828	122
TopBP-DL	.806		.781	.848	121
XGB-Score	.806				84
T-Bind	.818		.767		120
BsN-Score::XARG	.816				117
K_{DEEP}				.820	158
RF::XR	.806				85
BgN-Score::XARG	.804				117
RI-Score	.803		.762	.815	100
RF-Score-v2	.803				88
RF-Score-v3	.803				94
RF::CyscoreVinaElem	.803				87
CScore	.801	.767			115
BRT::XAR	.801				85
FFT-BP	.800			.747	119
SFCscore^{RF}	.779				86
B2Bscore		.746			95
RF-Score	.776	.728			79,115
SVM::XAR	.773				85
ID-Score	.753				107
CSM-Lig	.751				123
kNN::XA	.740				85
MARS::XAR	.710				85
RF@ML			.704		98
Pafnucy			.700	.780	159
BRT@ML			.694		98
kNN@ML			.672		98
MLR::XA	.689				85
$\Delta_{vina}RF_{20}$	.686		.732	.816	64,101
X-Score[c]	.644	.604	.614	.631	59,60,64,115

[a] In most cases, only the model with the best performance is listed in the table. But in some assessment publications, some top-ranked SFs with different ML methods are all enumerated.

[b] In most cases, the PDBbind core set was used as the test set, and the remaining part of the refined set that excludes the same complexes from the core set was used as the training set. Although some datasets such as v2013 and v2014 have the same core set, the difference in the training set can also lead to completely different results. So, only the most widely adopted PDBbind versions are included in the table, including v2007, v2009, v2013, and v2016.

[c] X-Score is a classical SF that shows the best scoring power according to the results of several assessments.

simple atomic features such as types, distances and partial charges as the input, which could not only simplify the input format but avoid the negative effects of some irrelevant features. Another universal characteristic of CNN-based methods was that most of them generated a 3D grid in advance to represent each complex. On the one hand, 3D structural data could be consequently described easily; on the other hand, features with the form of 3D grid could also be enabled to be compatible with the

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

TABLE 5 Screening power (AUC or $EF_{2\%}$) of different machine learning-based scoring functions on the Directory of Useful Decoys (DUD) or Database of Useful Decoys-Enhanced (DUD-E) dataset

		DUD		DUD-E		
Docking software	Rescoring method	AUC	$EF_{2\%}$	AUC	$EF_{2\%}$	Reference
Vina	/	0.70[a]				132
	NNScore	0.78				
	NNScore2.0	0.76				
Autodock4	/	0.60	5.6			116
	DDFA	0.74	9.6			
Vina	/	0.64[a]	7.1			
	DDFA	0.75	10.3			
RosettaLigand	/	0.65	6.2			
	DDFA	0.76	9.7			
Combining Vina, Autodock4 and RosettaLigand	/	0.70	7.4			
	DDFA	0.77	10.3			
Dock6.6	/	0.48	5.3			153
	DeepVS	0.74	5.9			
Vina	/	0.62	6.0			
	DeepVS	0.81	6.6			
Vina	/			0.74	8.2	99
	RF-Score-v3			0.67	4.47	
	RF-Score-VS			0.84	24.37	
Dock3.6	/			0.77	12.04	
	RF-Score-v3			0.71	5.96	
	RF-Score-VS			0.83	25.22	
Dock6.6	/			0.61	3.05	
	RF-Score-v3			0.66	3.75	
	RF-Score-VS			0.80	22.07	
Vina	/			0.716		90
	RF-Score			0.622		
	NNScore			0.584		
	3D-CNN			0.868		
Vina	/			0.703	7.135	91
	Baseline CNN			0.862	19.724	
	DenseFS			0.917	28.408	
Vina	/	0.64	6.9			121
	TopVS	0.84	9.5			

[a] The same strategy in different publications may yield a different result. This may result from the influence of other factors such as ligand and protein preparation, docking parameters, and dataset partition.

CNN just as that in image recognition. The introduction of other subdisciplines of mathematics to feature representation also leads in some new directions. For example, TopBP, TNet-BP, and T-bind introduced algebraic topology approaches for the description of biomolecular structures, and EI-Score used differential geometry to handle protein–ligand complexes.[120–122,157] The final triable strategy lies on the innovation of parameterization approach. A stark example was the design of $\Delta_{\text{vina}}\text{RF}_{20}$[101] that parameterized a correction term to improve the performance of the classical SFs, while conventionally the total binding affinities were used for training.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

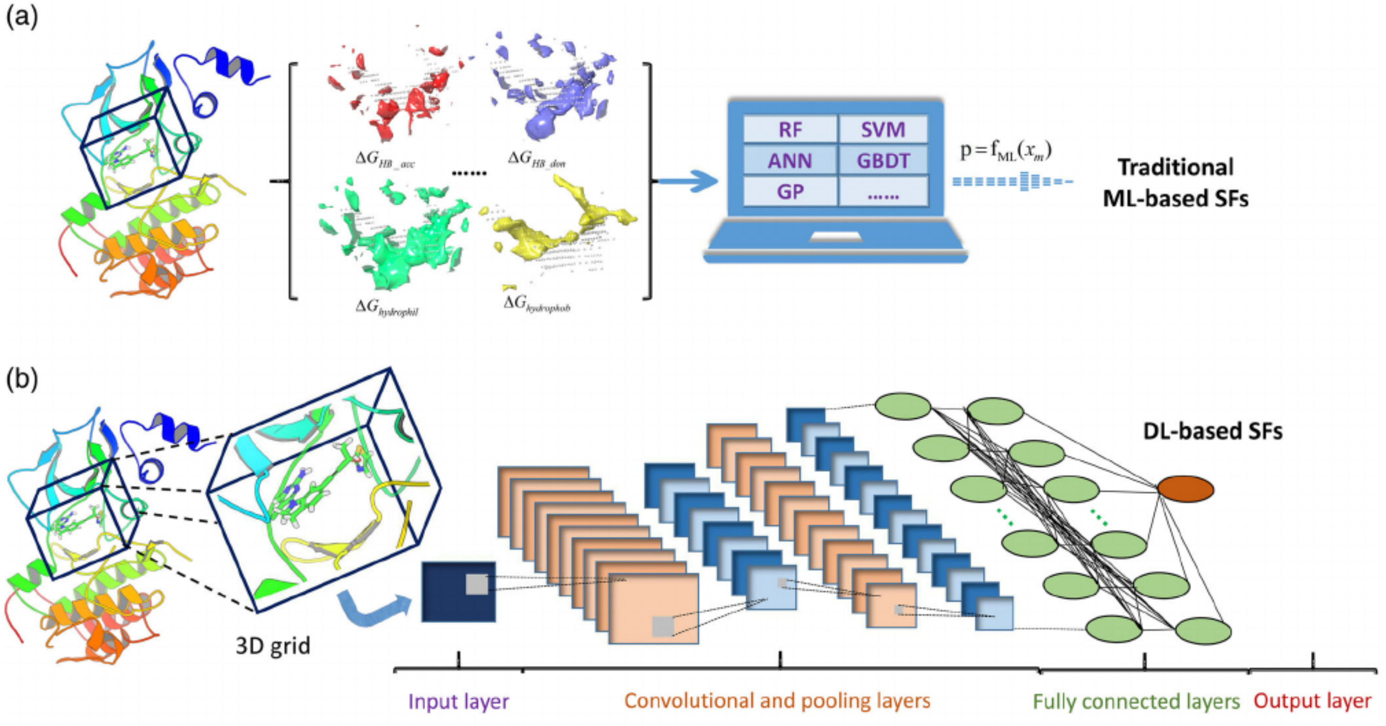

FIGURE 2 Typical workflows for the development of (a) traditional machine learning-based scoring functions (SFs) and (b) deep learning-based SFs

Nevertheless, there are also several defects or limitations urgently needed to be addressed. First of all, a benchmark specially designed for ML-based SFs demands prompt construction. CASF is well-performed for the assessment of classical SFs but it is not suitable enough for ML-based methods, because some external factors beyond the method itself such as the selection of training set and validation method may also affect the results a lot. Given the potential overfitting caused by highly structural or sequence similarity between the training and test sets, it is necessary to build more extensive external test sets for further validation. As for VS, another tough issue is the handling of decoys. In conventional DUD/DUD-E, decoys are generated with similar physical properties but dissimilar topology as active compounds. But a recent study found that due to the similarity of the artificial decoys for different targets, the models based on even some pure ligand-based descriptors could also bear excellent discrimination capabilities.[155] Therefore, to solve the above problems, some recently developed platforms such as Open Drug Discovery Toolkit (ODDT)[165] and MoleculeNet[143] may be utilized for reference, and a comprehensive assessment towards existing ML-based SFs with the newly developed benchmark is also required. Secondly, only few ML-based SFs have been integrated into the popular docking tools, and most of them just tend to be used as rescoring tools,[132,164] in which classical SFs are still irreplaceable so that reasonable poses can be generated first. So how to fully realize the potential of ML-based SFs to make them not just designed for rescoring is also an important direction in the future. Thirdly, a universal problem existing in ML and especially DL methods is the lack of interpretability. These models are usually treated as black boxes, whose internals are not so easy to be interpreted by humans.[167] In terms of ML-based SFs, a directly relevant concern is how to visualize the interpretation of individual protein–ligand complexes by ML models, which can in turn, not only guide medicinal chemistry optimization but also further inform the training and further improvement of the model. Several recently proposed ML-based SFs have taken the visualization capability of a model into consideration,[90,91,168] but obviously there is still a long way to go to realize highly efficient visualization.

In addition to the contents discussed above, the major outlook for the future still lies on the direct exploitation of novel and potent ML-based SFs, where state-of-the-art ML models and innovative feature representation methods should be mainly focused on. As more and more high-quality experimental data tends to be accessible publically, DL methods will gradually replace traditional ML approaches to become the mainstream technologies. Besides, with the rapid progress of computer science and artificial intelligence, an increasing number of novel DL methods will be accordingly developed and thereby fetched to be used for SF development. As for feature representation, other branch fields of cheminformatics such as Quantitative Structure–Activity Relationship[169] and Proteochemometric[170] modeling may provide sufficient resources. Finally, without a

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

doubt, we believe that the continuous improvement in ML-based SFs can surely facilitate the progress of drug design and discovery, and we hope this review will provide valuable insight for future research.

ACKNOWLEDGMENTS

This study was supported by the National Key R&D Program of China (2016YFA0501701, 2016YFB0201700), the National Science Foundation of China (81603031, 21575128, 81773632), and Zhejiang Provincial Natural Science Foundation of China (LZ19H300001).

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

ORCID

Tingjun Hou https://orcid.org/0000-0001-7227-2580

REFERENCES

1. Hughes JP, Rees S, Kalindjian SB, Philpott KL. Principles of early drug discovery. *Br J Pharmacol*. 2011;*162*:1239–1249.
2. Jorgensen WL. Efficient drug lead discovery and optimization. *Acc Chem Res*. 2009;*42*:724–733.
3. Clark DE. What has virtual screening ever done for drug discovery? *Expert Opin Drug Discovery*. 2008;*3*:841–851.
4. Scior T, Bender A, Tresadern G, et al. Recognizing pitfalls in virtual screening: A critical review. *J Chem Inf Model*. 2012;*52*:867–881.
5. Schneider G. Virtual screening: An endless staircase? *Nat Rev Drug Discov*. 2010;*9*:273–276.
6. Sliwoski G, Kothiwale S, Meiler J, Lowe EW Jr. Computational methods in drug discovery. *Pharmacol Rev*. 2014;*66*:334–395.
7. Lyne PD. Structure-based virtual screening: An overview. *Drug Discov Today*. 2002;*7*:1047–1055.
8. Cheng T, Li Q, Zhou Z, Wang Y, Bryant SH. Structure-based virtual screening for drug discovery: A problem-centric review. *AAPS J*. 2012; *14*:133–141.
9. Meng X-Y, Zhang H-X, Mezei M, Cui M. Molecular docking: A powerful approach for structure-based drug discovery. *Curr Comput Aided Drug Des*. 2011;*7*:146–157.
10. Chen Y-C. Beware of docking! *Trends Pharmacol Sci*. 2015;*36*:78–95.
11. Pagadala NS, Syed K, Tuszynski J. Software for molecular docking: A review. *Biophys Rev*. 2017;*9*:91–102.
12. Erickson BJ, Korfiatis P, Akkus Z, Kline TL. Machine learning for medical imaging(1). *Radiographics*. 2017;*37*:505–515.
13. Henglin M, Stein G, Hushcha PV, Snoek J, Wiltschko AB, Cheng S. Machine learning approaches in cardiovascular imaging. *Circ Cardiovasc Imaging*. 2017;*10*:e005614.
14. Ho HK, Zhang L, Ramamohanarao K, Martin S. A survey of machine learning methods for secondary and supersecondary protein structure prediction. *Methods Mol Biol*. 2013;*932*:87–106.
15. Jiang Q, Jin X, Lee S-J, Yao S. Protein secondary structure prediction: A survey of the state of the art. *J Mol Graph Model*. 2017;*76*:379–402.
16. Li J, Zheng S, Chen B, Butte AJ, Swamidass SJ, Lu Z. A survey of current trends in computational drug repositioning. *Brief Bioinform*. 2016; *17*:2–12.
17. Vanhaelen Q, Mamoshina P, Aliper AM, et al. Design of efficient computational workflows for in silico drug repurposing. *Drug Discov Today*. 2017;*22*:210–222.
18. Lavecchia A. Machine-learning approaches in drug discovery: Methods and applications. *Drug Discov Today*. 2015;*20*:318–331.
19. Lo Y-C, Rensi SE, Torng W, Altman RB. Machine learning in chemoinformatics and drug discovery. *Drug Discov Today*. 2018;*23*: 1538–1546.
20. Lima AN, Philot EA, Goulart Trossini GH, Barbour Scott LP, Maltarollo VG, Honorio KM. Use of machine learning approaches for novel drug discovery. *Expert Opin Drug Discovery*. 2016;*11*:225–239.
21. Vodrahalli K, Bhowmik AK. 3D computer vision based on machine learning with deep neural networks: A review. *J Soc Inf Disp*. 2017;*25*: 676–694.
22. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E. Deep learning for computer vision: A brief review. *Comput Intell Neurosci*. 2018;7068349.
23. Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process Mag*. 2012;*29*:82–97.
24. Silver D, Huang A, Maddison CJ, et al. Mastering the game of go with deep neural networks and tree search. *Nature*. 2016;*529*:484–489.
25. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag*. 2018; *13*:55–75.
26. Zhang L, Tan J, Han D, Zhu H. From machine learning to deep learning: Progress in machine intelligence for rational drug discovery. *Drug Discov Today*. 2017;*22*:1680–1685.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

27. Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T. The rise of deep learning in drug discovery. *Drug Discov Today*. 2018;*23*: 1241–1250.
28. Ghasemi F, Mehridehnavi A, Perez-Garrido A, Perez-Sanchez H. Neural network and deep-learning algorithms used in QSAR studies: Merits and drawbacks. *Drug Discov Today*. 2018;*23*:1784–1790.
29. Baskin II, Winkler D, Tetko IV. A renaissance of neural networks in drug discovery. *Expert Opin Drug Discovery*. 2016;*11*:785–795.
30. Rifaioglu AS, Atas H, Martin MJ, Cetin-Atalay R, Atalay V, Dogan T. Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases. *Brief Bioinform*. 2018.
31. Panteleev J, Gao H, Jia L. Recent applications of machine learning in medicinal chemistry. *Bioorg Med Chem Lett*. 2018;*28*:2807–2815.
32. Jing Y, Bian Y, Hu Z, Wang L, X-QS X. Deep learning for drug design: An artificial intelligence paradigm for drug discovery in the big data era. *AAPS J*. 2018;*20*:58.
33. Hessler G, Baringhaus K-H. Artificial intelligence in drug design. *Molecules*. 2018;*23*:2520.
34. Dana D, Gadhiya SV, St Surin LG, et al. Deep learning in drug discovery and medicine; scratching the surface. *Molecules*. 2018;*23*:2384.
35. Ain QU, Aleksandrova A, Roessler FD, Ballester PJ. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *WIREs Comput Molec Sci*. 2015;*5*:405–424.
36. Khamis MA, Gomaa W, Ahmed WF. Machine learning in computational docking. *Artif Intell Med*. 2015;*63*:135–152.
37. Kitchen DB, Decornez H, Furr JR, Bajorath J. Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nat Rev Drug Discov*. 2004;*3*:935–949.
38. Ferreira LG, dos Santos RN, Oliva G, Andricopulo AD. Molecular docking and structure-based drug design strategies. *Molecules*. 2015;*20*: 13384–13421.
39. Ewing TJA, Makino S, Skillman AG, Kuntz ID. DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases. *J Comput Aided Mol Des*. 2001;*15*:411–428.
40. Jones G, Willett P, Glen RC, Leach AR, Taylor R. Development and validation of a genetic algorithm for flexible docking. *J Mol Biol*. 1997; *267*:727–748.
41. Venkatachalam CM, Jiang X, Oldfield T, Waldman M. LigandFit: A novel method for the shape-directed rapid docking of ligands to protein active sites. *J Mol Graph Model*. 2003;*21*:289–307.
42. Guedes IA, Pereira FSS, Dardenne LE. Empirical scoring functions for structure-based virtual screening: Applications, critical aspects, and challenges. *Front Pharmacol*. 2018;*9*:1089.
43. Trott O, Olson AJ. Software news and update AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem*. 2010;*31*:455–461.
44. Friesner RA, Banks JL, Murphy RB, et al. Glide: A new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J Med Chem*. 2004;*47*:1739–1749.
45. Halgren TA, Murphy RB, Friesner RA, et al. Glide: A new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J Med Chem*. 2004;*47*:1750–1759.
46. Eldridge MD, Murray CW, Auton TR, Paolini GV, Mee RP. Empirical scoring functions. 1. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *J Comput Aided Mol Des*. 1997;*11*:425–445.
47. Wang RX, Lai LH, Wang SM. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J Comput Aided Mol Des*. 2002;*16*:11–26.
48. Gohlke H, Klebe G. Statistical potentials and scoring functions applied to protein–ligand binding. *Curr Opin Struct Biol*. 2001;*11*:231–235.
49. Muegge I, Martin YC. A general and fast scoring function for protein–ligand interactions: A simplified potential approach. *J Med Chem*. 1999;*42*:791–804.
50. Velec HFG, Gohlke H, Klebe G. DrugScore(CSD)-knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction. *J Med Chem*. 2005;*48*:6296–6303.
51. Mooij WTM, Verdonk ML. General and targeted statistical potentials for protein–ligand interactions. *Proteins Struct Funct Bioinf*. 2005;*61*: 272–287.
52. Baum B, Muley L, Smolinski M, Heine A, Hangauer D, Klebe G. Non-additivity of functional group contributions in protein–ligand binding: A comprehensive study by crystallography and isothermal titration calorimetry. *J Mol Biol*. 2010;*397*:1042–1054.
53. Breiman L. Random forests. *Machine Learning*. 2001;*45*:5–32.
54. Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995;*20*:273–297.
55. Vapnik VN. An overview of statistical learning theory. *IEEE Trans Neural Netw*. 1999;*10*:988–999.
56. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;*323*:533–536.
57. Seifert MHJ. Targeted scoring functions for virtual screening. *Drug Discov Today*. 2009;*14*:562–569.
58. Wang RX, Fang XL, Lu YP, Wang SM. The PDBbind database: Collection of binding affinities for protein–ligand complexes with known three-dimensional structures. *J Med Chem*. 2004;*47*:2977–2980.
59. Cheng T, Li X, Li Y, Liu Z, Wang R. Comparative assessment of scoring functions on a diverse test set. *J Chem Inf Model*. 2009;*49*: 1079–1093.
60. Li Y, Han L, Liu Z, Wang R. Comparative assessment of scoring functions on an updated benchmark: 2. Evaluation methods and general results. *J Chem Inf Model*. 2014;*54*:1717–1736.
61. Li Y, Liu Z, Li J, et al. Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the test set. *J Chem Inf Model*. 2014;*54*:1700–1716.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

62. Liu Z, Li Y, Han L, et al. PDB-wide collection of binding data: Current status of the PDBbind database. *Bioinformatics*. 2015;*31*:405–412.
63. Liu Z, Su M, Han L, et al. Forging the basis for developing protein–ligand interaction scoring functions. *Acc Chem Res*. 2017;*50*:302–309.
64. Su M, Yang Q, Du Y, et al. Comparative assessment of scoring functions: The CASF-2016 update. *J Chem Inf Model*. 2018;*59*:895–913.
65. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res*. 2000;*28*:235–242.
66. Huang N, Shoichet BK, Irwin JJ. Benchmarking sets for molecular docking. *J Med Chem*. 2006;*49*:6789–6801.
67. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of useful decoys, enhanced (DUD-E): Better ligands and decoys for better benchmarking. *J Med Chem*. 2012;*55*:6582–6594.
68. Rohrer SG, Baumann K. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *J Chem Inf Model*. 2009;*49*:169–184.
69. Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res*. 2016;*44*:D1045–D1053.
70. Wang Y, Bryant SH, Cheng T, et al. PubChem BioAssay: 2017 update. *Nucleic Acids Res*. 2017;*45*:D955–D963.
71. Gaulton A, Hersey A, Nowotka M, et al. The ChEMBL database in 2017. *Nucleic Acids Res*. 2017;*45*:D945–D954.
72. Smith RD, Dunbar JB Jr, Ung PM-U, et al. CSAR benchmark exercise of 2010: Combined evaluation across all submitted scoring functions. *J Chem Inf Model*. 2011;*51*:2115–2131.
73. Dunbar JB Jr, Smith RD, Damm-Ganamet KL, et al. CSAR data set release 2012: Ligands, affinities, complexes, and docking decoys. *J Chem Inf Model*. 2013;*53*:1842–1852.
74. Carlson HA, Smith RD, Damm-Ganamet KL, et al. CSAR 2014: A benchmark exercise using unpublished data from Pharma. *J Chem Inf Model*. 2016;*56*:1063–1077.
75. Gathiaka S, Liu S, Chiu M, et al. D3R grand challenge 2015: Evaluation of protein–ligand pose and affinity predictions. *J Comput Aided Mol Des*. 2016;*30*:651–668.
76. Gaieb Z, Liu S, Gathiaka S, et al. D3R grand challenge 2: Blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. *J Comput Aided Mol Des*. 2018;*32*:1–20.
77. Wu Z, Ramsundar B, Feinberg EN, et al. MoleculeNet: A benchmark for molecular machine learning. *Chem Sci*. 2018;*9*:513–530.
78. Gabel J, Desaphy J, Rognan D. Beware of machine learning-based scoring functions-on the danger of developing black boxes. *J Chem Inf Model*. 2014;*54*:2807–2815.
79. Ballester PJ, Mitchell JBO. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*. 2010;*26*:1169–1175.
80. Kramer C, Gedeck P. Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets. *J Chem Inf Model*. 2010;*50*:1961–1969.
81. Li Y, Yang J. Structural and sequence similarity makes a significant impact on machine-learning-based scoring functions for protein–ligand interactions. *J Chem Inf Model*. 2017;*57*:1007–1012.
82. Ballester PJ, Mitchell JBO. Comments on "Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets": Significance for the validation of scoring functions. *J Chem Inf Model*. 2011;*51*:1739–1741.
83. Li H, Peng J, Leung Y, et al. The impact of protein structure and sequence similarity on the accuracy of machine-learning scoring functions for binding affinity prediction. *Biomolecules*. 2018;*8*:E12.
84. Li H, Peng J, Sidorov P, et al. Classical scoring functions for docking are unable to exploit large volumes of structural and interaction data. *Bioinformatics*. 2019.
85. Ashtawy HM, Mahapatra NR. A comparative assessment of predictive accuracies of conventional and machine learning scoring functions for protein–ligand binding affinity prediction. *IEEE/ACM Trans Comput Biol Bioinform*. 2015;*12*:335–347.
86. Zilian D, Sotriffer CA. SFCscore(RF): A random forest-based scoring function for improved affinity prediction of protein–ligand complexes. *J Chem Inf Model*. 2013;*53*:1923–1933.
87. Li H, Leung K-S, Wong M-H, Ballester PJ. Substituting random forest for multiple linear regression improves binding affinity prediction of scoring functions: Cyscore as a case study. *BMC Bioinform*. 2014;*15*:291.
88. Ballester PJ, Schreyer A, Blundell TL. Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity? *J Chem Inf Model*. 2014;*54*:944–955.
89. Gomes J, Ramsundar B, Feinberg EN, Pande VS. Atomic convolutional networks for predicting protein–ligand binding affinity. arXiv e-prints 2017 [cited 30 Mar 2017]. Available from: https://ui.adsabs.harvard.edu/#abs/2017arXiv170310603G.
90. Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR. Protein–ligand scoring with convolutional neural networks. *J Chem Inf Model*. 2017;*57*: 942–957.
91. Imrie F, Bradley AR, van der Schaar M, Deane CM. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. *J Chem Inf Model*. 2018;*58*:2319–2330.
92. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics*. 2007;*23*:2507–2517.
93. Colwell LJ. Statistical and machine learning approaches to predicting protein–ligand interactions. *Curr Opin Struct Biol*. 2018;*49*:123–128.
94. Li H, Leung K-S, Wong M-H, Ballester PJ. Improving AutoDock Vina using random forest: The growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. *Mol Inform*. 2015;*34*:115–126.
95. Liu Q, Kwoh CK, Li J. Binding affinity prediction for protein–ligand complexes based on beta contacts and B factor. *J Chem Inf Model*. 2013; *53*:3076–3085.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

96. Ashtawy HM, Mahapatra NR. A comparative assessment of ranking accuracies of conventional and machine-learning-based scoring functions for protein–ligand binding affinity prediction. *IEEE/ACM Trans Comput Biol Bioinform*. 2012;*9*:1301–1313.
97. Ashtawy HM, Mahapatra NR. Machine-learning scoring functions for identifying native poses of ligands docked to known and novel proteins. *BMC Bioinform*. 2015;*16*:S3.
98. Khamis MA, Gomaa W. Comparative assessment of machine-learning scoring functions on PDBbind 2013. *Eng Appl Artif Intel*. 2015;*45*: 136–151.
99. Wojcikowski M, Ballester PJ, Siedlecki P. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci Rep*. 2017;*7*:46710.
100. Nguyen DD, Xiao T, Wang M, Wei G-W. Rigidity strengthening: A mechanism for protein–ligand binding. *J Chem Inf Model*. 2017;*57*: 1715–1721.
101. Wang C, Zhang Y. Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *J Comput Chem*. 2017;*38*:169–177.
102. Yasuo N, Sekijima M. Improved method of structure-based virtual screening via interaction-energy-based learning. *J Chem Inf Model*. 2019; *59*:1050–1061.
103. Li L, Wang B, Meroueh SO. Support vector regression scoring of receptor–ligand complexes for rank-ordering and virtual screening of chemical libraries. *J Chem Inf Model*. 2011;*51*:2132–2138.
104. Li L, Khanna M, Jo I, et al. Target-specific support vector machine scoring in structure-based virtual screening: Computational validation, in vitro testing in kinases, and effects on lung cancer cell proliferation. *J Chem Inf Model*. 2011;*51*:755–759.
105. Xu D, Meroueh SO. Effect of binding pose and modeled structures on SVMGen and GlideScore enrichment of chemical libraries. *J Chem Inf Model*. 2016;*56*:1139–1151.
106. Koppisetty CAK, Frank M, Kemp GJL, Nyholm P-G. Computation of binding energies including their enthalpy and entropy components for protein–ligand complexes using support vector machines. *J Chem Inf Model*. 2013;*53*:2559–2570.
107. Li G-B, Yang L-L, Wang W-J, Li L-L, Yang S-Y. ID-score: A new empirical scoring function based on a comprehensive set of descriptors related to protein–ligand interactions. *J Chem Inf Model*. 2013;*53*:592–600.
108. Das S, Krein MP, Breneman CM. Binding affinity prediction with property-encoded shape distribution signatures. *J Chem Inf Model*. 2010; *50*:298–308.
109. Sato T, Honma T, Yokoyama S. Combining machine learning and pharmacophore-based interaction fingerprint for in silico screening. *J Chem Inf Model*. 2010;*50*:170–185.
110. Ding B, Wang J, Li N, Wang W. Characterization of small molecule binding. I. Accurate identification of strong inhibitors in virtual screening. *J Chem Inf Model*. 2013;*53*:114–122.
111. Yan Y, Wang W, Sun Z, Zhang JZH, Ji C. Protein–ligand empirical interaction components for virtual screening. *J Chem Inf Model*. 2017;*57*: 1793–1806.
112. Nogueira MS, Koch O. The development of target-specific machine learning models as scoring functions for docking-based target prediction. *J Chem Inf Model*. 2019;*59*:1238–1252.
113. Durrant JD, McCammon JA. NNScore: A neural-network-based scoring function for the characterization of protein–ligand complexes. *J Chem Inf Model*. 2010;*50*:1865–1871.
114. Durrant JD, McCammon JA. NNScore 2.0: A neural-network receptor-ligand scoring function. *J Chem Inf Model*. 2011;*51*:2897–2903.
115. Ouyang X, Handoko SD, Kwoh CK. Cscore: A simple yet effective scoring function for protein–ligand binding affinity prediction using modified CMAC learning architecture. *J Bioinform Comput Biol*. 2011;*9*:1–14.
116. Arciniega M, Lange OF. Improvement of virtual screening results by docking data feature analysis. *J Chem Inf Model*. 2014;*54*:1401–1411.
117. Ashtawy HM, Mahapatra NR. BgN-score and BsN-score: Bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein–ligand complexes. *BMC Bioinform*. 2015;*16*:S8.
118. Ashtawy HM, Mahapatra NR. Task-specific scoring functions for predicting ligand binding poses and affinity and for screening enrichment. *J Chem Inf Model*. 2018;*58*:119–133.
119. Wang B, Zhao Z, Nguyen DD, Wei G-W. Feature functional theory-binding predictor (FFT-BP) for the blind prediction of binding free energies. *Theor Chem Acc*. 2017;*136*:1–22.
120. Cang Z, Wei G-W. Integration of element specific persistent homology and machine learning for protein–ligand binding affinity prediction. *Int J Numer Method Biomed Eng*. 2018;*34*:e2914.
121. Cang Z, Mu L, Wei G-W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Comput Biol*. 2018;*14*:e1005929.
122. Nguyen DD, Wei G-W. DG-GL: Differential geometry-based geometric learning of molecular datasets. *Int J Numer Method Biomed Eng*. 2019;*35*:e3179.
123. Pires DEV, Ascher DB. CSM-lig: A web server for assessing and comparing protein–small molecule affinities. *Nucleic Acids Res*. 2016;*44*: W557–W561.
124. Schreyer A, Blundell T. CREDO: A protein–ligand interaction database for drug discovery. *Chem Biol Drug Des*. 2009;*73*:157–167.
125. Sotriffer CA, Sanschagrin P, Matter H, Klebe G. SFCscore: Scoring functions for affinity prediction of protein–ligand complexes. *Proteins Struct Funct Bioinf*. 2008;*73*:395–419.
126. Brereton RG, Lloyd GR. Support vector machines for classification and regression. *Analyst*. 2010;*135*:230–267.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

127. Amini A, Shrimpton PJ, Muggleton SH, Sternberg MJE. A general approach for developing system-specific functions to score protein–ligand docked complexes using support vector inductive logic programming. *Proteins Struct Funct Bioinf*. 2007;*69*:823–831.
128. Kinnings SL, Liu N, Tonge PJ, Jackson RM, Xie L, Bourne PE. A machine learning-based method to improve docking scoring functions and its application to drug repurposing. *J Chem Inf Model*. 2011;*51*:408–419.
129. Sun H, Pan P, Tian S, et al. Constructing and validating high-performance MIEC-SVM models in virtual screening for kinases: A better way for actives discovery. *Sci Rep*. 2016;*6*:24817.
130. Rosenblatt F. The perceptron—A probabilistic model for information-storage and organization in the brain. *Psychol Rev*. 1958;*65*:386–408.
131. Durrant JD, McCammon JA. BINANA: A novel algorithm for ligand-binding characterization. *J Mol Graph Model*. 2011;*29*:888–893.
132. Durrant JD, Friedman AJ, Rogers KE, McCammon JA. Comparing neural-network scoring functions and the state of the art: Applications to common library screening. *J Chem Inf Model*. 2013;*53*:1726–1735.
133. Durrant JD, Carlson KE, Martin TA, et al. Neural-network scoring functions identify structurally novel estrogen-receptor ligands. *J Chem Inf Model*. 2015;*55*:1953–1961.
134. Lindert S, Zhu W, Liu Y-L, Pang R, Oldfield E, McCammon JA. Farnesyl diphosphate synthase inhibitors from in silico screening. *Chem Biol Drug Des*. 2013;*81*:742–748.
135. Durrant JD, Amaro RE. Machine-learning techniques applied to antibacterial drug discovery. *Chem Biol Drug Des*. 2015;*85*:14–21.
136. Skaff DA, McWhorter WJ, Geisbrecht BV, Wyckoff GJ, Miziorko HM. Inhibition of bacterial mevalonate diphosphate decarboxylase by eriochrome compounds. *Arch Biochem Biophys*. 2015;*566*:1–6.
137. Buryska T, Daniel L, Kunka A, Brezovsky J, Damborsky J, Prokop Z. Discovery of novel haloalkane dehalogenase inhibitors. *Appl Environ Microbiol*. 2016;*82*:1958–1965.
138. Friedman JH. Greedy function approximation: A gradient boosting machine. *Ann Stat*. 2001;*29*:1189–1232.
139. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. arXiv e-prints 2016 [cited 9 Mar 2016]. Available from: https://ui.adsabs.harvard.edu/#abs/2016arXiv160302754C.
140. Adam-Bourdarios C, Cowan G, Germain-Renaud C, Guyon I, Kegl B, Rousseau D. The Higgs machine learning challenge. *Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics*, Okinawa, Japan, Vol. 664; 2015.
141. Volkovs M, Yu GW, Poutanen T. Content-based neighbor models for cold start in recommender systems. *Proceedings of the Recommender Systems Challenge Workshop 2017*, Como, Italy; 2017.
142. Sandulescu V, Chiru M. Predicting the future relevance of research institutions—The winning solution of the KDD Cup 2016. arXiv e-prints 2016 [cited 9 Sep 2016]. Available from: https://ui.adsabs.harvard.edu/#abs/2016arXiv160902728S.
143. Ashtawy HM, Mahapatra NR. Descriptor Data Bank (DDB): A cloud platform for multiperspective modeling of protein–ligand interactions. *J Chem Inf Model*. 2018;*58*:134–147.
144. Seeger M. Gaussian processes for machine learning. *Int J Neural Syst*. 2004;*14*:69–106.
145. Min S, Lee B, Yoon S. Deep learning in bioinformatics. *Brief Bioinform*. 2017;*18*:851–869.
146. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;*15*:1929–1958.
147. Dahl GE, Sainath TN, Hinton GE. Improving deep neural networks for Lvcsr using rectified linear units and dropout. *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013; p. 8609–8613.
148. Sainath TN, Vinyals O, Senior A, Sak H. Convolutional long short-term memory, fully connected deep neural networks. *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, Australia, 2015; p. 4580–4584.
149. Lawrence S, Giles CL, Tsoi AC, Back AD. Face recognition: A convolutional neural-network approach. *IEEE Trans Neural Netw*. 1997;*8*:98–113.
150. Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput*. 1989;*1*:270–280.
151. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;*9*:1735–1780.
152. Wallach I, Dzamba M, Heifets A. AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv e-prints 2015 [cited 10 Oct 2015]. Available from: https://ui.adsabs.harvard.edu/#abs/2015arXiv151002855W.
153. Pereira JC, Caffarena ER, dos Santos CN. Boosting docking-based virtual screening with deep learning. *J Chem Inf Model*. 2016;*56*:2495–2506.
154. Ragoza M, Turner L, Koes DR. Ligand pose optimization with atomic grid-based convolutional neural networks. arXiv e-prints 2017 [cited 20 Oct 2017]. Available from: https://ui.adsabs.harvard.edu/#abs/2017arXiv171007400R.
155. Gonczarek A, Tomczak JM, Zareba S, Kaczmar J, Dabrowski P, Walczak MJ. Interaction prediction in structure-based virtual screening using deep learning. *Comput Biol Med*. 2018;*100*:253–258.
156. Feinberg EN, Sur D, Wu Z, et al. PotentialNet for molecular property prediction. *ACS Central Sci*. 2018;*4*:1520–1530.
157. Cang Z, Wei G. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput Biol*. 2017;*13*:e1005690.
158. Jimenez J, Skalic M, Martinez-Rosell G, De Fabritiis G. K-DEEP: Protein–ligand absolute binding affinity prediction via 3D-convolutional neural networks. *J Chem Inf Model*. 2018;*58*:287–296.
159. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*. 2018;*34*:3666–3674.
160. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. *30th IEEE Conference on Computer Vision and Pattern Recognition*. New York: IEEE, 2017; p. 2261–2269.

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].

161. Rogers D, Hahn M. Extended-connectivity fingerprints. *J Chem Inf Model*. 2010;*50*:742–754.

162. Simoes RS, Maltarollo VG, Oliveira PR, Honorio KM. Transfer and multi-task learning in QSAR modeling: Advances and challenges. *Front Pharmacol*. 2018;*9*:74.

163. Sosnin S, Vashurina M, Withnall M, Karpov P, Fedorov M, Tetko IV. A survey of multi-task learning methods in chemoinformatics. *Mol Inform*. 2019;*38*:e1800108.

164. Li HJ, Leung KS, Ballester PJ, Wong MH. Istar: A web platform for large-scale protein–ligand docking. *PLoS One*. 2014;*9*:e85678.

165. Wojcikowski M, Zielenkiewicz P, Siedlecki P. Open drug discovery toolkit (ODDT): A new open-source player in the drug discovery field. *J Cheminform*. 2015;*7*:26.

166. Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. Molecular graph convolutions: Moving beyond fingerprints. *J Comput Aided Mol Des*. 2016;*30*:595–608.

167. Guidotti R, Monreale A, Ruggieri S, Turin F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. *ACM Comput Surv*. 2019;*51*:93.

168. Hochuli J, Helbling A, Skaist T, Ragoza M, Koes DR. Visualizing convolutional neural network protein–ligand scoring. *J Mol Graph Model*. 2018;*84*:96–108.

169. Polishchuk P. Interpretation of quantitative structure activity relationship models: Past, present, and future. *J Chem Inf Model*. 2017;*57*:2618–2639.

170. Qiu TY, Qiu JX, Feng J, et al. The recent progress in proteochemometric modelling: Focusing on target descriptors, cross-term descriptors and application scope. *Brief Bioinform*. 2017;*18*:125–136.

How to cite this article: Shen C, Ding J, Wang Z, Cao D, Ding X, Hou T. From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. *WIREs Comput Mol Sci*. 2020;10:e1429. https://doi.org/10.1002/wcms.1429

Printed by [Wiley Online Library - 060.114.118.175 - /doi/epdf/10.1002/wcms.1429] at [12/08/2020].