

Article

Identifying Potential Machine Learning Algorithms for the Simulation of Binding Affinities to Molecularly Imprinted Polymers

Joseph W. Lowdon ^{1,*} , Hikaru Ishikura ², Malene K. Kvernenes ², Manlio Caldara ¹, Thomas J. Cleij ¹ , Bart van Grinsven ¹ , Kasper Eersels ¹  and Hanne Diliën ¹ 

- ¹ Sensor Engineering Department, Faculty of Science and Engineering, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; m.caldara@maastrichtuniversity.nl (M.C.); thomas.cleij@maastrichtuniversity.nl (T.J.C.); bart.vangrinsven@maastrichtuniversity.nl (B.v.G.); kasper.eersels@maastrichtuniversity.nl (K.E.); hanne.dilien@maastrichtuniversity.nl (H.D.)
- ² Maastricht Science Programme, Faculty of Science and Engineering, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; h.ishikura@student.maastrichtuniversity.nl (H.I.); m.kvernenes@student.maastrichtuniversity.nl (M.K.K.)
- * Correspondence: joe.lowdon@maastrichtuniversity.nl

Abstract: Molecularly imprinted polymers (MIPs) are synthetic receptors engineered towards the selective binding of a target molecule; however, the manner in which MIPs interact with other molecules is of great importance. Being able to rapidly analyze the binding of potential molecular interferences and determine the selectivity of a MIP can be a long tedious task, being time- and resource-intensive. Identifying computational models capable of reliably predicting and reporting the binding of molecular species is therefore of immense value in both a research and commercial setting. This research therefore sets focus on comparing the use of machine learning algorithms (multitask regressor, graph convolution, weave model, DAG model, and inception) to predict the binding of various molecular species to a MIP designed towards 2-methoxyphenidine. To this end, each algorithm was “trained” with an experimental dataset, teaching the algorithms the structures and binding affinities of various molecular species at varying concentrations. A validation experiment was then conducted for each algorithm, comparing experimental values to predicted values and facilitating the assessment of each approach by a direct comparison of the metrics. The research culminates in the construction of binding isotherms for each species, directly comparing experimental vs. predicted values and identifying the approach that best emulates the real-world data.

Keywords: molecularly imprinted polymers; artificial intelligence; neural networking; simulation



Citation: Lowdon, J.W.; Ishikura, H.; Kvernenes, M.K.; Caldara, M.; Cleij, T.J.; van Grinsven, B.; Eersels, K.; Diliën, H. Identifying Potential Machine Learning Algorithms for the Simulation of Binding Affinities to Molecularly Imprinted Polymers. *Computation* **2021**, *9*, 103. <https://doi.org/10.3390/computation9100103>

Academic Editor: Karlheinz Schwarz

Received: 13 August 2021

Accepted: 23 September 2021

Published: 24 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent decades, advancements in the field of biosensing have precipitated the development of point-of-care (PoC) technologies for the rapid analysis of a whole host of molecules [1,2]. The main driving force behind these marvels lies in the advancement of the receptor materials utilized in the transduction of a biological/chemical target into a tangible output signal [3]. Molecularly imprinted polymers (MIPs) are a primary example of a developed synthetic receptor material, consisting of a highly crosslinked polymeric network with selective nanocavities engineered for the binding of an analyte distributed throughout [4,5]. These synthetic alternatives to biological recognition elements demonstrate enhanced resistance to extreme physical conditions and can operate in a wide range of scenarios, which is highlighted in the increasing number of applications that have been linked to these synthetic wonders [6].

Traditionally, MIPs have been developed through a time-consuming trial-and-error methodology consisting of altering the chemical compositions and ratios of the monomer

and crosslinker species to achieve desired results [7,8]. The same tedium exists for determining the performance of a MIP, having to expose it to countless molecular moieties to truly determine the selective capabilities of the polymer [9,10]. Due to the nature of PoC analysis, this testing process is vital, though traditional approaches have proven costly and protracted. The potential to lessen these constrictors does exist and lies within the realms of data science, more commonly referred to as computational chemistry. Many chemical processes have already benefited from the implementation of diverse algorithms, revolutionizing the fields of retrosynthesis, reaction optimization, and drug design [11–13]. Computational methods have also been applied to molecularly imprinted polymers, though they focus on the compositional nature rather than the binding of various analytes [14–17].

One computational tool that can aid the development of MIPs is machine learning, and in particular the use of neural networks (NN). This kind of machine learning methodology is most commonly applied in the fields of chemo-informatics, enabling the prediction of the physical and chemical properties of molecules, or even how a compound is likely to react [18–20]. As more datasets on molecules have become available, the use of NN has become increasingly more popular, with this approach offering more reliable chemical predictions [21,22]. This success can be derived from the different NN architectures possible, enabling chemical information to be layered and processed in numerous ways that allow new trends and relationships to be perceived [23]. The dataset and desired outcomes of analysis dictate which model is selected, though the selection may not have an obvious choice and requires investigation.

A wide range of NN architectures has been documented, including Directed Acyclic Graphs (DAG), Spatial Graph convolution, Multitask Regressors, and image classification models, to name just a few [24–27]. Each has a unique manner in which it processes data, though all require molecular structures and other variables to be preprocessed prior to analysis. This preprocessing is known as featurizing, a process that can also affect the way the model selected interprets the data [28,29]. Molecular structures are featurized by converting them into Simplified Molecular Input Line Entry Specification (SMILES), transforming a molecular structure into a line of text that the models can interpret [30]. The manner in which the rest of the data (including the SMILES) is featurized depends on the model selected, adding to the complexity of model selection. All in all, the best method of selecting a model is a trial and error approach, attempting different models and selecting the best performing as a basis to be built upon.

Due to the complex nature of NN, knowledge in the field is critical for success. As the field of neural networking has expanded, an increasing number of research groups in many different areas has adopted these algorithms for data analysis, making the demand for experts in NN higher than ever before [31]. This increase in demand has led to the development of deep learning frameworks such as DeepChem that offer a host of deep learning architectures that can be modified and used with little experience [32–34]. DeepChem is a python-based framework that specifically targets chemistry, offering deep learning algorithms tailored towards the analysis of molecular datasets, thus making it a perfect platform for building and testing molecular NN models for the analysis and prediction of MIP binding behavior.

The presented research sets focus on using this framework to evaluate which model and featurizer simulates best the binding of specific substrates to a MIP when compared to real world data. To this end, four different featurizers were utilized in the modelling of this data: the RDKitDescriptors, weave featurizer, ConvMolFeaturizer, and the SmilesToImage featurizers. The featurized datasets were then coupled to three different NN models: the Directed Acyclic Graph (DAG) model, Graph Convolution model, and multitask regressor. These models were trained and tested with a combination of data from previous publications and data collected specifically for the purpose of this publication.

2. Dataset Selection

Over the years, MIPs have been applied to the detection of many different small molecules, demonstrating how they can be used as synthetic receptors in many different fields [35]. A lot of data therefore exists in regard to different compositions of MIPs with a corresponding template. The data however lack depth, with specific MIP binding data generally only extending towards the template molecule and a few analogues/interferants [36]. Finding datasets that provide a sufficient amount of molecular species binding to a MIP is therefore a tricky task.

An area where extensive binding data is of high importance is the field of drug screening. This area has seen a surge of activity in recent years with the “umbrella” of illicit substances being expanded to encompass new chemical architectures beyond classical drugs of abuse [37]. This extended prohibition now covers compounds coined New Psychoactive Substances (NPS), being compounds that are synthetically modified variants of classical compounds that contain a psychoactive pharmacophore with an overall different chemical architecture [38]. A prime example of such a compound is 2-methoxphenidine (2-MXP). Being akin to the classical compounds ketamine and phencyclidine (PCP), 2-MXP acts on the NMDA receptors and has dissociative effects on the user [39]. Other positional isomers of the compound exist, and current preliminary test methodologies can easily mistake the compound for a whole variety of compounds, including caffeine, which is not controlled [40]. It is therefore key to have binding data that covers a large array of molecules for sensors designed for these kinds of compounds, as false positives/negatives have become a greater issue as chemical architectures stray away from the norm.

With this in mind, the data selected for the training and testing of the generated neural network models are derived from the NPS field. With the research groups’ prior experience in this area, there is already MIP based data covering a range of chemical structures that can be used for training and test the models. The data were compiled from multiple NPS-based papers [41,42] and unpublished results to produced datasets for the training of the models. This training set was randomly selected from the data collected from the sources, introducing the models to a variety of structures in a bid to increase the models’ recognition of a diverse range of molecules. Each molecule introduced has a corresponding SMILES, initial concentration of substrate (C_i), amount of substrate bound to the MIP at equilibrium (S_b), and amount of substrate remaining in solution at equilibrium (C_f). The same was true for the testing dataset, though the number of molecules for this was dramatically reduced, as the bulk of the molecular data was used for the training of the models.

The following research therefore outlines the methods used to train and test models based on specific algorithms, evaluating these models using statistical techniques (R^2 , Pearson R^2 , and MAE score), alongside comparing the predicted values to those of the experimental data. Overall, the research endeavored to identify which of the algorithms are the most promising towards modelling MIP binding affinities based on limited datasets.

3. Modelling Methodology

3.1. Hardware and Software

All computational modelling and calculations were conducted on a desktop system (components and technical specifications: CPU—AMD Ryzen 7 3700x, MSI B550-A Pro motherboard ATX, 16 Gb DDR 4, 1 TB M.2 SSD, GPU—GeForce RTX 3070 OC 8 G, 850 W power supply) installed with Windows 10. Models and featurizers from DeepChem framework (with TensorFlow GPU enabled) were used for model training and testing, utilizing Jupyter Notebook to run the calculations, and data analysis was conducted with the NumPy package. The collected data was then plotted with OriginPro (2020 edition).

3.2. Datasets

All the data used in the training and testing of the computational models were gathered from previous publications or are unpublished data gathered with the distinct purpose of further characterizing the binding affinities of the selected MIP [41,42]. The MIP binding

affinities for 2-MXP, 3-MXP, 4-MXP, caffeine, 2-MEPE, 3-MEPE, 4-MEPE, diphenidine, ephenidine, phenol red, crystal violet, malachite green, basic blue, paracetamol, aspirin, methyl orange, and sucrose were derived from these papers. The corresponding NIP data were collected to compliment these datapoints and to provide reference data for the computational models for comparison.

For the “training” of the NN models a dataset consisting of binding capacities (S_b) and free concentrations (C_f) at defined initial concentrations (C_i) for a variety of molecular species towards the target MIP was presented (see Supplementary Information Table S1). The molecular data in this training set were represented by the corresponding SMILES of each molecule, with this field being the primary featurized component of the data. Each model was then trained towards the prediction of the S_b and corresponding C_f at defined C_i values (0.1–0.5 mM). The same process was also completed for the corresponding MIP data to act as a reference (see Supplementary Information Table S2).

Three molecules were selected for the testing of the models, consisting of 3-methoxyphenidine, diphenidine, and 3-methoxyephendine. The molecules selected were structural analogues of molecules in the training set, and were expected provide a valid metric of determining whether the models were performing sufficiently. As with the training set, the S_b and C_f values at defined C_i were determined and compared to the collected experimental values (see Supplementary Information Table S3). The same process was also completed for the corresponding MIP data to act as a reference (see Supplementary Information Table S4).

3.3. Directed Acyclic Graph (DAG) Model

An initial step of preprocessing the molecular data into a DAG compatible format was required, converting molecular structures into nodes/vertices (atoms) and edges (bonds) (Figure 1). This was achieved by processing the initial data frames with a DAG transformer, producing said vertices and edges for each molecule. As the graphs generated were acyclic, the orientation and direction of the graphs had been defined; therefore, all vertices were cycled through, designating each one vertex in turn the root of the graph. All other remaining vertices and edges were then direct towards the selected root, producing the DAG (example given in Figure 1, for the 2-methoxyphenidine). As the data were built around the binding of simple molecules, it is feasible that a DAG can be generated for each atom within the molecule within a reasonable timeframe. In short, if a molecule has N vertices correlated with N atoms it will produce N DAG [43].

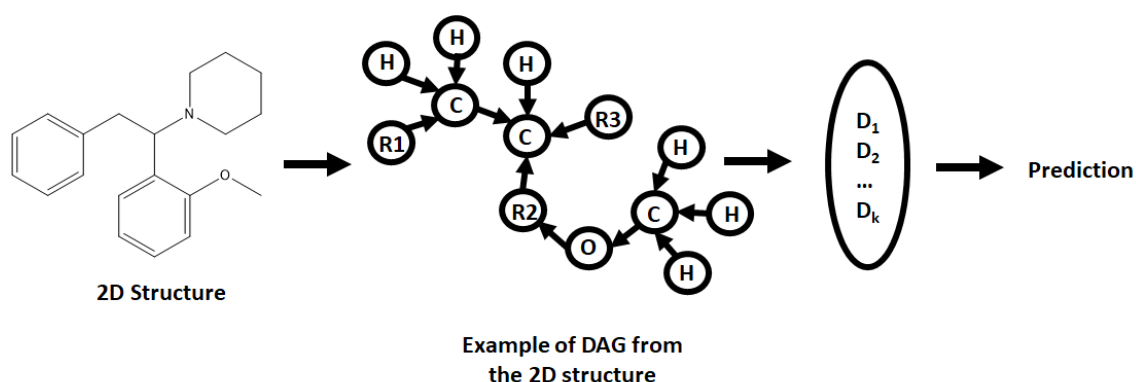


Figure 1. Examples of how a molecule (2-methoxyphenidine) can be converted into a directed acyclic graph (DAG), and then how the sum of different root versions of the DAG are used for a prediction.

Each node (v) has an associated contextual vector ($G_{v,k}$) indexed by k (the associated DAG). $G_{v,k}$ correlates information about a corresponding atom and its neighbors (or parent vertices ($pa^1_{v,k}, \dots, pa^n_{v,k}$)), with an input vector i and a function M^G that implements recursion (Equation (1)) based on other parameters selected for the neural network [43]. The sum from each of the generated DAGs are then combined, producing a DAG-RNN

(DAG-recursive neural network), relating all the information and producing an output value.

$$G_{v,k} = M^G \left(i_v, G_{pa^1_{[v,k]}}, \dots, G_{pa^n_{[v,k]}} \right) \quad (1)$$

This model was constructed using the model builder in the DeepChem framework, training the model dataset using the default parameters settings (max_atoms = 50, n_atom_feat = 75, n_graph_feat = 30, n_outputs = 30, layer_sizes = [100], layer_sizes_gather = [100], dropout = none, model = regression, n_classes = 2), before using the basic model for parameter optimization. Hyperparameter optimization was conducted using a grid approach, determining the optimum model across a range of parameters (Appendix A) basing the optimum values off the generated R^2 and Pearson R^2 metrics. These parameters were analyzed across a range (10–1200) of epochs (iterations) to determine the optimum number of iterations to avoid over training of the model. The optimized model was then used to predict the S_b and corresponding C_f values at defined C_i values for the testing dataset. These values were then plotted alongside their experimental counterparts to produce a comparative binding isotherm.

3.4. Graph Convolution Model

Similar to DAG, the graph convolution model has the ability to represent a molecular structure as vertices (atoms) and edges (bonds) (Figure 2). The main difference lies in how the layers of the model are processed, with the model taking the product of translated functions (known as convolution) to extrapolate extra information or to find more complex patterns within the given data. At the lower levels of the network the same filter (convolution) is applied to neighboring atoms and bonds, with the information learned being combined after several steps generating a global pool that represents larger chunks of the network. The process can then be repeated for the higher levels, finding more relationships within the data and building a model of greater complexity [44].

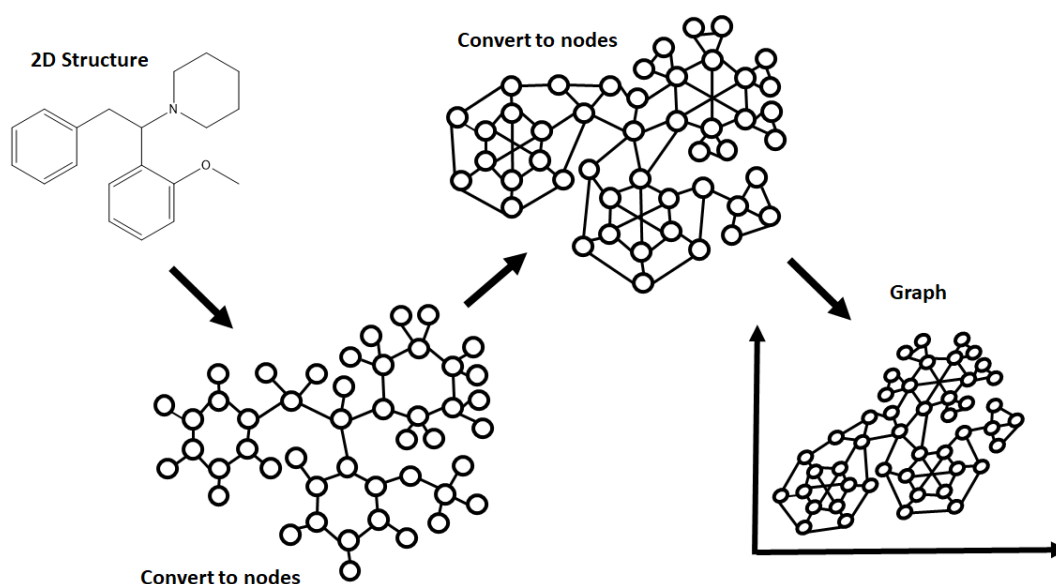


Figure 2. The concept behind the graph convolution model, stemming from the conversion of a model compound (2-methoxyphenidine) into nodes that are ready for convolution and graphing.

This model was constructed using the model builder in the DeepChem framework, training the model dataset using the default parameters settings (number_atom_features = 75, graph_conv_layers = [64, 64], dense_layer_size = 128, dropout = none, model = regression, n_classes = 2), before using the basic model for parameter optimization. Hyperparameter optimization was conducted using a grid approach, determining the optimum model across a range of parameters (Appendix A) basing the optimum values of the generated R^2 and

Pearson R^2 metrics. These parameters were analyzed across a range (10–1200) of epochs (iterations) to determine the optimum number of iterations to avoid over training of the model. The optimized model was then used to predict the S_b and corresponding C_f values at defined C_i values for the testing dataset. These values were then plotted alongside their experimental counterparts to produce a comparative binding isotherm.

3.5. Graph Convolution Weave Model

A different variant of the graph convolution model exists, consolidating primitive operations between atoms and their corresponding layers into a single module. Thus, rather than finding correlations in big chunks of data, the model finds links between smaller chunks going back and forth in nature. The back and forth nature of the learned relationships giving rise to the coined name “weave” convolution. The degree to which the data is weaved can easily be controlled by limiting the amount of “back and forth” data processing that occurs, with over iteration of this process having negative effects on the learned data. This methodology therefore has appeal, as it has the potential to produce more informative representations of the original input, and enables the option of stacking weave modules with limited atom pairs to produce more in-depth models [45].

As with the other models, the model was constructed using the DeepChem framework model builder, training the model data using the default parameter settings ($n_atom_feat = 75$, $n_pair_feat = 14$, $n_hidden = 50$, $n_graph_feat = 128$, $n_weave = 2$, $fully_connected_layer_sizes = [2000, 100]$, $conv_weight_init_stddevs = 0.03$, $weight_init_stddevs = 0.01$, $dropouts = 0.25$, $mode = regression$) before using the basic model for parameter optimization. Hyperparameter optimization was conducted using a grid approach, determining the optimum model across a range of parameters (Appendix A) basing the optimum values off the generated R^2 and Pearson R^2 metrics. These parameters were analyzed across a range (10–1200) of epochs (iterations) to determine the optimum number of iterations to avoid over training of the model. The optimized model was then used to predict the S_b and corresponding C_f values at defined C_i values for the testing dataset. These values were then plotted alongside their experimental counterparts to produce a comparative binding isotherm.

3.6. Multitask Regressor Model

A multitask regressor approach would enable the training of multiple targets with the same input features, e.g., to classify/regress a molecule on both its 2D chemical structure and associated chemical properties. This means the model has more than one input that the model handles simultaneously and can be used to predict the regression product (Figure 3) [46]. This could be ideal for the analysis of molecular interactions with MIPs, as there are multiple chemical properties that can be factored into the model, producing more detailed and comprehensive predictions.

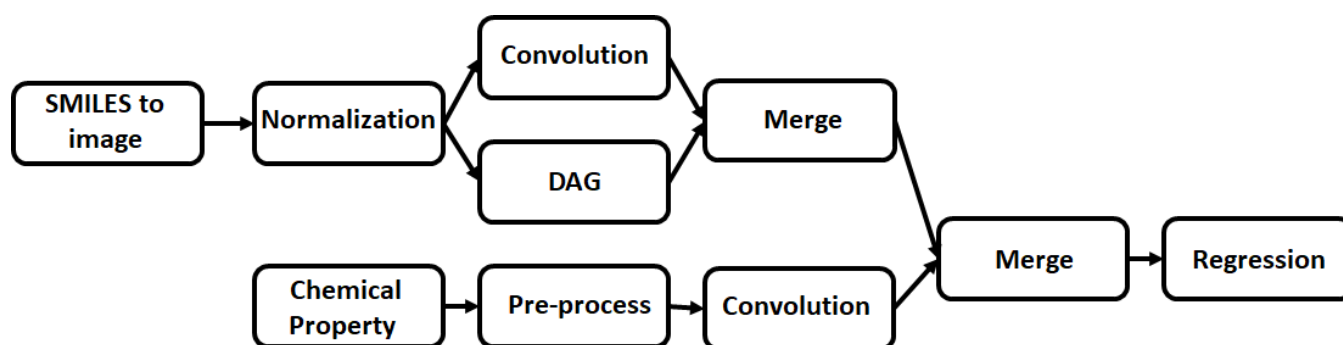


Figure 3. The concept behind the multitask regressor algorithm, simultaneously predicting values based off different chemical properties and then merging these values to generate an overall numerical prediction.

To this end, the RdKit descriptors were selected that could take the molecular SMILE and compute a list of chemical descriptors that are relevant to the selected molecule,

e.g., molecular weights, number of valence electrons, partial charge, size, etc. [47]. This information was then used to featurize the experimental binding data to train the regressor model. To this end the DeepChem framework model builder was utilized for model training and testing using default parameters ($n_features = 208$, $layer_sizes = [1000, 1000, 1000]$, $dropout = 0.2$, $learn_rate = 0.0001$) before using this base model for optimization. Hyperparameter optimization was conducted using a grid approach, determining the optimum model across a range of parameters (Appendix A) basing the optimum values off the generated R^2 and Pearson R^2 metrics. These parameters were analyzed across a range (10–1200) of epochs (iterations) to determine the optimum number of iterations to avoid over training of the model. The optimized model was then used to predict the S_b and corresponding C_f values at defined C_i values for the testing dataset. These values were then plotted alongside their experimental counterparts to produce a comparative binding isotherm.

3.7. ChemCception Model

An alternative method of data analysis is possible by using the ChemCception model of molecular analysis where a 2D structure of the molecule is generated from the SMILES, which in turn is mapped onto an array and used to train a neural network. The method essentially uses the 2D image of the molecule and breaks it down into a pixel grid which then has the measured chemical properties associated with it and a neural network generated that is trained towards the likeness of molecular structures (Figure 4). This simplistic approach bypasses a lot of other chemical characteristics that the other methods employ, though this technique has proven to be effective in other areas of cheminformatics.

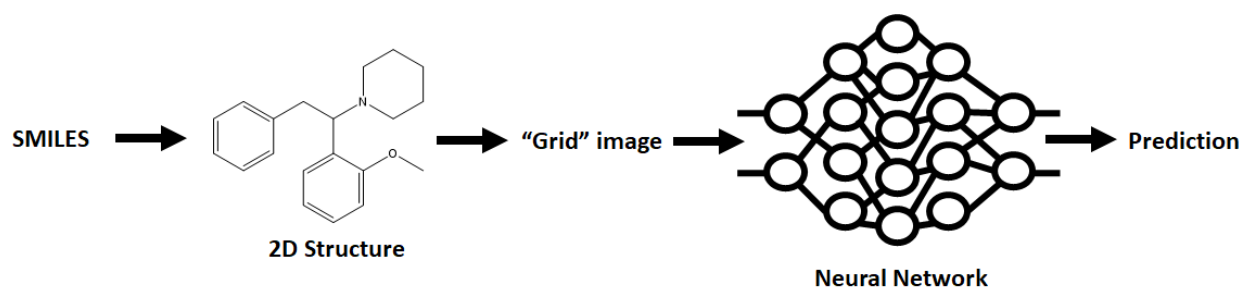


Figure 4. The concept behind the ChemCception methodology of generating a neural network that relates the 2D structure of a molecule to a trained chemical property.

As with the other algorithms, the model was constructed using the DeepChem framework model builder, training the model using the default parameter settings ($img_spec = std$, $img_size = 80$, $base_filters = 16$, $inception_blocks = dict$, $n_classes = 2$, $mode = regression$) before using the basic model for parameter optimization. As the model relies upon the transforming of an image into a neural network, a SMILES to image featurizer was utilized prior to the training and testing of the model data. Hyperparameter optimization was conducted using a grid approach, determining the optimum model across a range of parameters (Appendix A) basing the optimum values off the generated R^2 and Pearson R^2 metrics. These parameters were analyzed across a range (10–1200) of epochs (iterations) to determine the optimum number of iterations to avoid over training of the model. The optimized model was then used to predict the S_b and corresponding C_f values at defined C_i values for the testing dataset. These values were then plotted alongside their experimental counterparts to produce a comparative binding isotherm.

4. Results and Discussion

4.1. Direct Acyclic Graph (DAG) Model (ConvMolFeaturizer)

After optimization of the DAG model, it was trained at each C_i value individually, recording the MAE score, R^2 , Pearson R^2 , and RMS for each of the trained models (Table 1).

This process was complete for both MIP and NIP datasets, enabling a critical review of the trained model to evaluate how well each model learned the given data.

Table 1. Analysis of the error and correlation of the trained datasets at a given C_i value.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.9989	0.9999	0.452	0.9999	0.9999	0.153
	0.2	0.9999	0.9999	0.0003	0.9998	0.9998	0.0005
	0.3	0.9999	0.9999	0.055	0.9999	0.9999	2.683
	0.4	0.9999	0.9999	0.001	0.9999	0.9999	0.000006
	0.5	0.9999	0.9999	0.001	0.9999	0.9999	0.00001
NIP	0.1	0.9999	0.9999	0.163	0.9999	0.9999	0.00003
	0.2	0.9999	0.9999	0.0004	0.9999	0.9999	0.000007
	0.3	0.9999	0.9999	0.006	0.9999	0.9999	0.000003
	0.4	0.9999	0.9999	0.001	0.9999	0.9999	0.000006
	0.5	0.9999	0.9999	0.002	0.9999	0.9999	0.000001

The metrics revealed that each of the models was sufficiently trained, having low RMS error across the full C_i range for both MIP and NIP while also demonstrating good positive correlation highlighted by the R^2 and Pearson R^2 . The MAE score measured the average magnitude of error in the models, with this value being sub 1 for all of the trained models. This process of analyzing the models was repeated for the trained data (Table 2), facilitating a comparison between the trained and tested data.

Table 2. Analysis of the error and correlation of the test datasets at a given C_i value.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.3342	0.4504	12.42	0.9776	0.9897	13.71
	0.2	0.6102	0.7569	6.809	0.8496	0.9027	0.019
	0.3	0.7720	0.8409	6.653	0.8321	0.8323	0.025
	0.4	0.5169	0.9568	12.56	0.9378	0.9689	0.023
	0.5	0.4813	0.9549	12.44	0.7832	0.8314	0.051
NIP	0.1	0.0164	0.0981	13.53	0.1304	0.1281	0.039
	0.2	0.2769	0.7313	14.72	0.6355	0.6853	0.022
	0.3	0.5885	0.8035	16.59	0.7138	0.7193	0.029
	0.4	0.3157	0.4226	16.20	0.4555	0.4863	0.066
	0.5	0.5671	0.9134	13.67	0.9247	0.9381	0.030

The predicted test data were plotted against the experimental data to produce the binding isotherms that correlate the amount of substrate bound to the MIP/NIP (S_b) against the free concentration in solution (C_f) (Figure 5). The plotted data revealed that the model was generally better at predicting values for the MIP verses the NIP, though this is to be expected as the MIP should have a degree of predictable specific binding where the NIP should only have non-specific interactions. It can be assumed that the prediction of non-specific interactions is harder to simulate than the specific interactions as they are somewhat more random in nature, and relies more upon the distribution of binding functionalities across the surface of the polymer rather than the distribution of nanocavities. Of the tested compounds, the predicted values for 3-MXP were the closest to the experimental values for both MIP and NIP, though the resemblance between the simulated and experimental values was not by any means perfect. Over the entire concentration range, the model over predicted the S_b values, whereas all the NIP values were under predicted apart from the fourth datapoint, which reflected the experimental values within a reasonable tolerance.

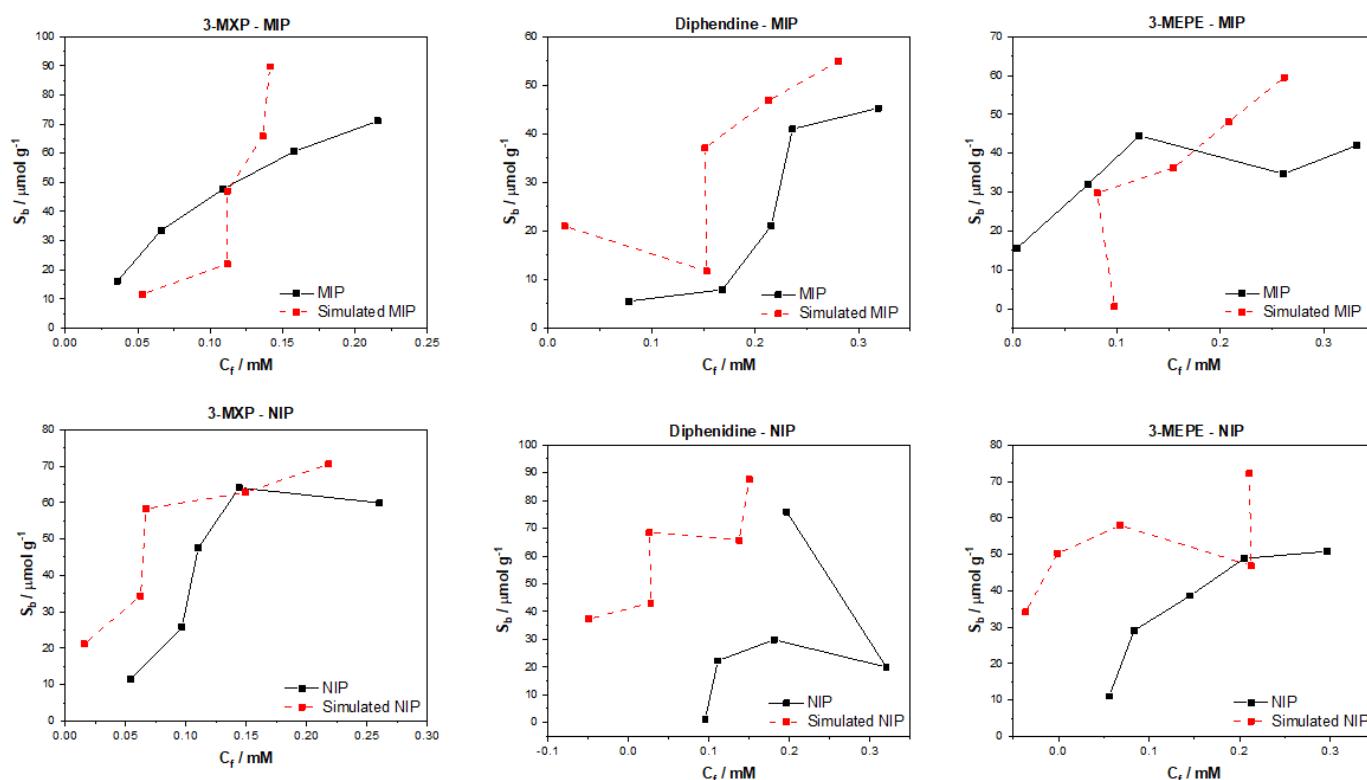


Figure 5. DAG simulated and experimental binding isotherms for MIP and NIP in regard to the binding of 3-MXP, diphenidine, and 3-MEPE.

The simulated data for 3-MEPE were shown to follow the same trend as observed with the 3-MXP datapoints, though the values simulated were further from the experimental values bar a couple of points for both MIP/NIP. The simulated values for the diphenidine MIP resembled the experimental values in trend, though again all the values seemed to be higher in comparison to the real-world values. The predicted values for the NIP were completely off, producing data that were unrecognizable from the experimental data. As the model interprets the SMILES as a series of nodes and edges that are directed towards a given root, the algorithm might not form deep enough relationships to truly simulate the binding the testing molecules. It therefore may be more beneficial to select an approach that builds deeper correlations at the lower layers and can find links between nodes/edges that are further away to produce more accurate predictions.

4.2. Multitask Regressor Model (RdKit Featurizer)

As with the previous model, a critical review of how well the model performed with the training and testing data was conducted by analyzing the correlation coefficients and MAE score. The training of the model appeared to go well with strong R^2 /Pearson R^2 values across the board for both MIP/NIP. The training data for the NIP, however, proved to show greater correlation than that of the MIP, though a downwards trend in the correlation values was visible as C_i increased (Table 3). The MAE score must also be taken into consideration to understand the true picture with this score being of a similar magnitude for both the imprinted and non-imprinted datasets. This suggests that the model responded in a similar manner to the training of both MIP/NIP. The interpretation of this score differed across the C_i range, as lower S_b and C_i values correlated with the lower initial concentrations. With this in mind, it is clear that the errors at the lower C_i values were more significant than those at higher concentrations. The MAE error in the MIP data was, however, shown to increase across the C_i range, though this trend was not necessarily true for NIP data. Thus, though there was acceptable correlation in the trained models, the error within these models could be considered higher than desired.

This analysis of the metrics suggests that the multitask regressor algorithm was learning from the data fed to it, though there was clearly some information missing that could lead to a higher degree of correlation and lower error in the generated models.

Table 3. Analysis of the error and correlation of the trained datasets at a given C_i value for the multitask regressor model.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.9157	0.9498	3.922	0.9701	0.9787	17.42
	0.2	0.6479	0.6791	5.258	0.3921	0.4843	0.191
	0.3	0.7215	0.8305	5.958	0.4944	0.7981	0.150
	0.4	0.8143	0.8289	7.392	0.6923	0.7769	0.209
	0.5	0.8450	0.8450	8.508	0.6272	0.8122	0.172
NIP	0.1	0.9521	0.9712	2.344	0.0110	0.0689	0.188
	0.2	0.9341	0.9375	4.073	0.8276	0.4445	0.236
	0.3	0.9211	0.9657	4.973	0.2176	0.6963	0.193
	0.4	0.9278	0.9503	4.316	0.6454	0.7515	0.196
	0.5	0.8557	0.8617	7.808	0.1117	0.8759	0.168

The same analysis conducted with the test data revealed that the trained model performed less well with unfamiliar data (Table 4). In general, there was no real trend to be seen within the statistics for either MIP or NIP, though the MIP data produced the lower MAE score, indicating a lesser average magnitude of error in the results. The MAE score was generally high across all the predictions, demonstrating that though there was a correlation in the datapoints, there was high error on average. The performance of the models did not seem to increase or decrease with increasing C_i , with the correlation proving to be independent for each differing free concentration. Overall, most coefficients had values greater than 0.6, suggesting that there was some correlation within the models. When comparing these values to the training of the models, it is clear that the test data performed less well. This is to be expected, as the models were responding to unfamiliar molecules, though the similarities in these molecules to the training set should be enough to stimulate a higher degree of positive correlation.

Table 4. Analysis of the error and correlation of the test datasets at a given C_i value for the multitask regressor model.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.5231	0.5501	10.40	0.8933	0.9805	22.25
	0.2	0.7157	0.8759	6.226	0.1207	0.2465	0.215
	0.3	0.6734	0.8242	9.228	0.4642	0.6449	0.183
	0.4	0.7678	0.8992	8.199	0.4164	0.9023	0.239
	0.5	0.2203	0.8302	18.30	0.8120	0.9565	0.191
NIP	0.1	0.4481	0.5374	11.46	0.3188	0.7790	0.1997
	0.2	0.5729	0.9330	11.20	0.2871	0.7003	0.2561
	0.3	0.6988	0.9093	7.539	0.0913	0.0462	0.225
	0.4	0.0157	0.4359	14.85	0.2873	0.8130	0.222
	0.5	0.6536	0.8784	12.92	0.2533	0.8427	0.221

Plotting of the simulated datapoints against the experimental values revealed that for the MIP, the predicted values were close to the actual values for 3-MXP and 3-MEPE (Figure 6). The deviations from the experimental values in these cases were slight, and the trend for each molecule was clearly present. This could also be said for diphenidine, though the predicted values were higher than the experimental values, and this effect seemed to be accentuated towards the higher C_f values. These were surprising results

considering how poor the model seemed to perform under statistical analysis. This said, the plotted results have the same level of uncertainty associated with them as the analysis of the models represented in Table 4, thus meaning that although they may appear accurate and a good representation of the experimental values, they still have the associated error and uncertainty. The generated NIP data in comparison were a lot poorer, with none of the simulated datasets being close to those of the experimental values. Again, this could be due to the nature of the seemingly non-specific nature of the binding towards the NIPs, which is hard to predict. The overall simulated datapoints' MAE score for both MIP and NIP is high, demonstrating a larger average error in the predicted values. The effect of this was lessened as C_i values increased, though not enough for the error to become negligible. The multitask regressor model itself seems better tailored for more predictable/organized data, as the model simultaneously attempted to analyze various facets of the molecule and its binding properties. When the binding was more random it can be thought that this approach was less effective due to the higher level of uncertainty in the actual binding of the molecules. This said, this model should shine with the limit dataset that was utilized in the training and testing of the model, though it appears that the model performs less well when compared to the other models critiqued in this manuscript.

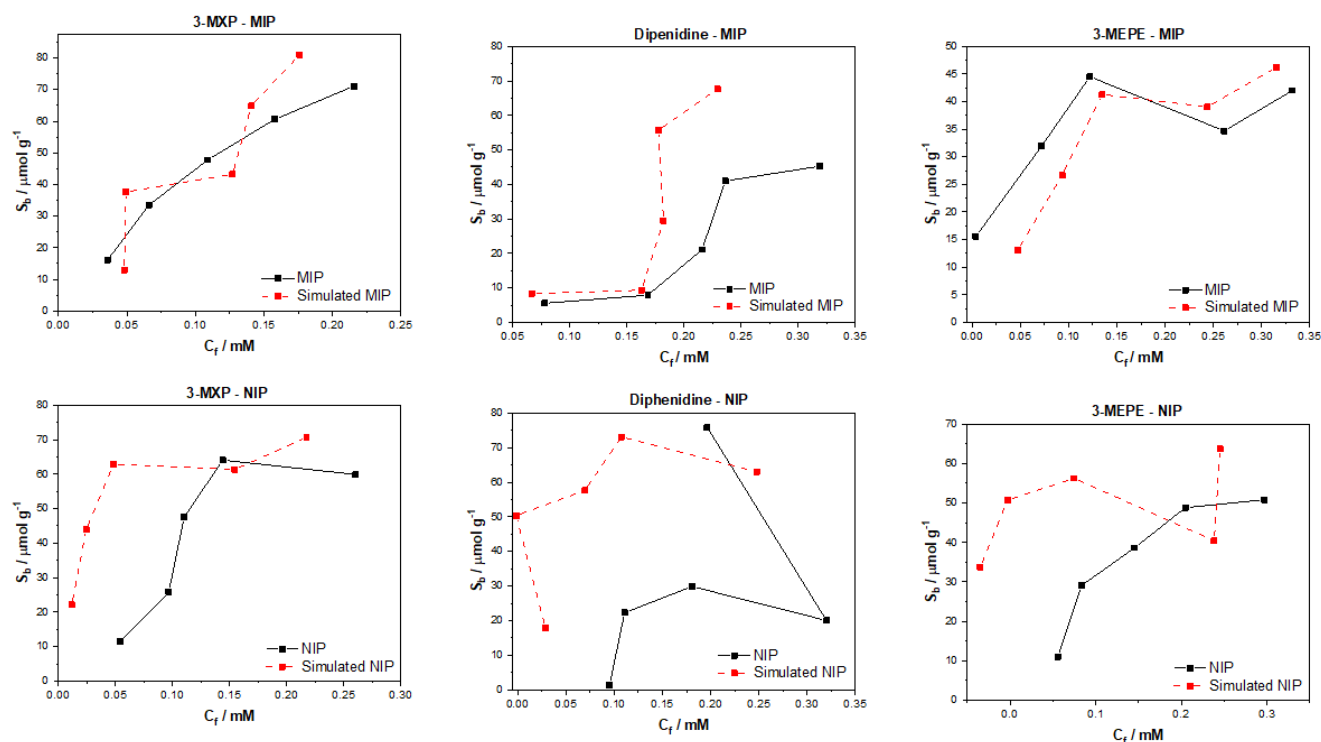


Figure 6. Multitask regressor with the RdKit featurizer simulated and experimental binding isotherms for MIP and NIP in regard to the binding of 3-MXP, diphenidine, and 3-MEPE.

4.3. Graph Convolution Model (ConvMolFeaturizer)

When critically analyzing the metrics associated with how well the MIP/NIP models were trained, it is clear that across the full C_i range, the Pearson R^2 , R^2 , and MAE scores were all excellent (Table 5). All of the trained models had correlation coefficients close to 1, indicating a strong positive correlation. The corresponding MAE scores complimented these statistics, proving to be relatively low and therefore demonstrating that there was a low average magnitude in error across the range of trained models. The MIP and NIP model training both had excellent statistical results, highlighting that it is possible to reliably train the graph convolution towards both types of data.

Table 5. Analysis of the error and correlation of the trained datasets at a given C_i value for the graph convolution model.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.9999	0.9999	0.019	0.7522	0.9133	29.92
	0.2	0.9999	0.9999	0.012	0.9579	0.9830	0.011
	0.3	0.9999	0.9999	0.055	0.9897	0.9929	0.009
	0.4	0.9996	0.9998	0.294	0.9904	0.9912	0.010
	0.5	0.9965	0.9988	1.047	0.9930	0.9978	0.011
NIP	0.1	0.9999	0.9999	0.040	0.9454	0.9573	0.011
	0.2	0.9999	0.9999	0.107	0.9942	0.9945	0.005
	0.3	0.9999	0.9999	0.096	0.9917	0.9923	0.009
	0.4	0.9996	0.9998	0.288	0.9942	0.9942	0.011
	0.5	0.9986	0.9994	0.708	0.9988	0.9989	0.004

This trend ended when the models were subjected to the test datasets, with the correlation coefficients and the MAE score for all samples being affected (Table 6). The correlation coefficients for both MIP and NIP were drastically reduced, with a lot of the models tending towards an R^2 of 0.6, indicating a weak correlation in the testing data. The associated MAE scores demonstrated a greater magnitude in error, highlighting how the model was overall less capable at statistically predicting test molecule values. This is in direct contrast to the trained data statistics above, with the models now under performing in comparison. The fact that this trend was true for both MIP and NIP suggests that the issue could stem from the limited datasets applied for testing, or the nature of the algorithm itself, though the method proved effective for the training of the models, which makes the limited test dataset the more likely cause of the issue.

Table 6. Analysis of the error and correlation of the test datasets at a given C_i value for the graph convolution model.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.2671	0.5283	7.634	0.3371	0.6829	13.43
	0.2	0.5478	0.8645	7.419	0.2248	0.6230	0.240
	0.3	0.6868	0.7385	8.685	0.2446	0.4595	0.308
	0.4	0.1513	0.6655	13.97	0.1544	0.0231	0.301
	0.5	0.4571	0.7282	20.20	0.0065	0.0062	0.345
NIP	0.1	0.4724	0.6098	8.818	0.5782	0.6395	0.251
	0.2	0.4251	0.9404	6.485	0.0563	0.0531	0.309
	0.3	0.6328	0.9408	8.824	0.0058	0.0013	0.442
	0.4	0.3422	0.3705	15.88	0.4495	0.7524	0.409
	0.5	0.5875	0.8289	13.05	0.1183	0.2355	0.348

When the trained models were used to generate a binding isotherm for 3-MXP, a clear likeness between the simulated and experimental data was observed for the MIP (Figure 7). The simulated NIP isotherm was, however, off for the lower concentrations but became more accurate as the higher concentrations were reached. The shape of the datapoints for the 3-MXP binding for MIP/NIP was accurate for the experimental data, suggesting the model understood the trend in the data, yet lacked information on how well the 3-MXP should bind to the NIP at lower concentrations. The model predicted the trend in the diphenidine MIP data, though greatly overestimated the S_b across the concentration range other than the lowest C_i (0.1 mM). When simulating the NIP for both diphenidine and 3-MEPE, the model struggled to replicate the experimental values, though it worked for the 3-MEPE interaction with the MIP at lower concentrations. Overall, the model performs poorly across each of the molecules apart from 3-MXP, with the trends in the simulated

datapoints somewhat resembling the experimental data. This demonstrates that the model has the capacity to learn trends well but needs more information if it is to replicate the S_b and C_f values more accurately. As the model is a convolution system, it should be considered modifying the reach of the convolution to further/closer atomic neighbors to see how this affects the model. Other physical parameters could be used in the testing of the model, though a greater dataset for analysis may prove more useful for a full evaluation of the model.

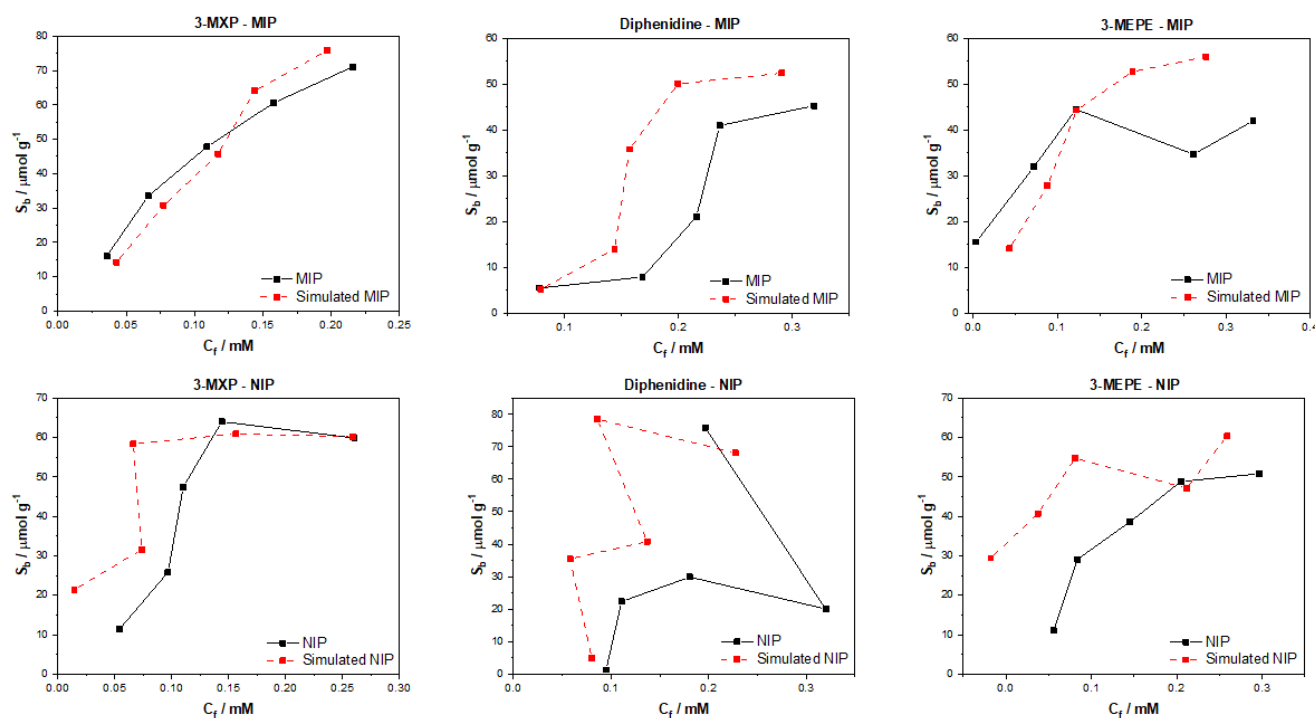


Figure 7. Graph convolution simulated and experimental binding isotherms for MIP and NIP in regard to the binding of 3-MXP, diphenidine, and 3-MEPE.

4.4. Graph Convolution Model (Weave Featurizer)

The graph convolution from the previous experiment was modified and featurized with a weave featurizer to identify if the alternative featurization would affect the training and testing of the graph convolution model. To enable a comparison, the same statistical analyses of the trained models were conducted (Table 7). The use of the featurizer was found to have a negative effect on the correlation coefficients and MAE score of the trained model across the entire C_i range. The coefficients appeared to decrease as the initial concentration increased for the MIP, with the MAE score showing a similar trend. The NIP did not appear to show the same trend, with values for the correlation coefficients seeming to be independent of the C_i . When comparing this to the graph convolution without the weave featurizer, it showed a decrease in training performance, though the correlation coefficients remained relatively high in comparison to some of the other models tested. The main feature that seemed to be affected was the MAE score, with values across both MIP and NIP data being increased by a magnitude of 10–100. This shows a dramatic increase in the average error of each model and therefore means that the trained models were less reliable when compared to the previous model. The function of the weave featurizer to interconnect layers back and forth may be the reason for the increase in the error, with greater uncertainty being introduced as layers are now linked in a non-linear fashion.

Table 7. Analysis of the error and correlation of the trained datasets at a given C_i value for the graph convolution model with weave featurizer.

	C_i	S_b			C_b		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.9676	0.9820	2.742	0.4011	0.7459	7.191
	0.2	0.9417	0.9610	2.431	0.7765	0.8678	0.261
	0.3	0.9707	0.9892	2.754	0.1896	0.1142	0.147
	0.4	0.9075	0.9488	5.645	0.3442	0.4641	0.366
	0.5	0.03925	0.1571	28.56	0.1752	0.5457	0.117
NIP	0.1	0.1066	0.9727	13.18	0.0204	0.0888	0.218
	0.2	0.9687	0.9813	3.362	0.0374	0.0371	0.109
	0.3	0.9493	0.9812	4.930	0.1697	0.1352	0.362
	0.4	0.3715	0.8553	16.88	0.0234	0.1610	0.7454
	0.5	0.7766	0.8838	13.14	0.4246	0.8027	0.327

This same affect can be seen in the test datasets for the models, with the average correlation coefficient tending towards 0.5 (Table 8). Though this is only a decrease of 0.1, it indicates that the tested data were harder to correlate and performed less well when compared to the pure graph convolution model without the featurizer. The associated MAE score for both MIP and NIP remained consistent with what was previously observed, though in comparison, the weave featurizer tended to produce the higher error overall. Again, with the training of the model, the reason for this likely lies with the nature of the featurizer used, and the back and forth facet that has been introduced to the model.

Table 8. Analysis of the error and correlation of the test datasets at a given C_i value for the graph convolution model with weave featurizer.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.6182	0.6357	9.725	0.8761	0.9591	3.591
	0.2	0.6114	0.9109	6.852	0.6249	0.7089	0.163
	0.3	0.5132	0.6333	9.845	0.3356	0.3952	0.162
	0.4	0.5371	0.9691	11.70	0.8301	0.9261	0.628
	0.5	0.05479	0.3989	15.28	0.7828	0.9562	0.049
NIP	0.1	0.1962	0.5291	6.680	0.2053	0.8194	0.338
	0.2	0.1284	0.8579	14.09	0.1876	0.6224	0.073
	0.3	0.3283	0.9219	12.05	0.7259	0.8882	0.501
	0.4	0.5666	0.1575	25.04	0.4727	0.7507	0.755
	0.5	0.5774	0.6071	10.10	0.6323	0.9036	0.450

Not surprisingly, these results rippled through to the generation of the binding isotherms for the test molecules (Figure 8). Again, 3MXP was the strongest of the molecules, providing a simulated binding isotherm for both MIP and NIP close to the experimental values. The values generated were similar to the experimental values, though the trend that was previously seen with the graph convolution was no longer apparent. Values for diphenidine and 3-MEPE were of a similar manner to the associated experimental MIP values, though the model seemed to over predict (diphenidine) or under predict (3-MEPE) by a considerable margin. When analyzing the NIP data, the values predicted were completely off and did not show signs of matching the trends in the experimental data. When coupling these binding isotherms with the statistical analysis above, the excess error and reduced correlation coefficients suggest that the poor performance of the model may stem from the weaker training and less successful testing of the model. Overall, it should be considered that the use of the weave featurizer is more problematic for the smaller datasets, and a larger dataset would be required to make full use of the “weave” functionality that is introduced.

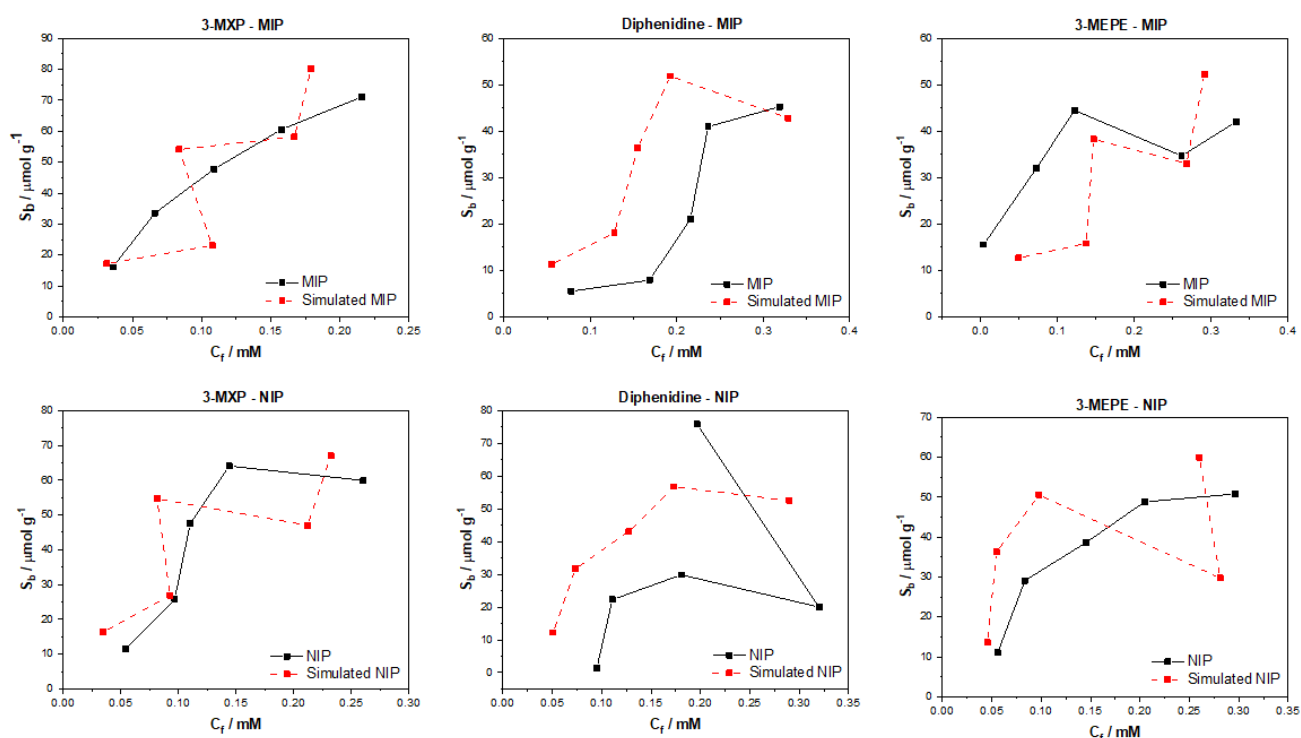


Figure 8. Graph convolution with a weave featurizer simulated and experimental binding isotherms for MIP and NIP in regard to the binding of 3-MXP, diphenidine, and 3-MEPE.

4.5. ChemCepion Model (SMILESToImageFeaturizer)

The ChemCepion model is the least similar to all the other approaches attempted, converting the molecular SMILES into a 2D image that is then divided into pixels and analyzed as a grid. This enables the direct comparison of the 2D chemical structures as the model operates on image comparison. This in theory should enable the rapid testing of other structures as the model focuses primarily on the similarities in the generated 2D images rather than solely focusing on associated chemical parameters and values between bonds and atoms. The trained model was therefore analyzed in the same manner as those previously, looking at the correlation coefficients and MAE score (Table 9). The trained models for MIP and NIP demonstrated highly positive correlations with corresponding low MAE scores, demonstrating that the models generated responded well to the training data and had little associated error. This is promising as it highlights how well the algorithm recognized the 2D structures when they were converted to pixel grids.

Table 9. Analysis of the error and correlation of the trained datasets at a given C_i value for the graph ChemCepion model with SMILESToImage featurizer.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.9999	0.9999	0.01638	0.9999	0.9999	0.075
	0.2	0.9997	0.9999	0.6011	0.9999	0.9999	0.000007
	0.3	0.9999	0.9999	0.4671	0.9999	0.9999	0.0000004
	0.4	0.9997	0.9999	0.06748	0.9999	0.9999	0.000001
	0.5	0.9999	0.9999	0.099	0.9999	0.9999	0.0000007
NIP	0.1	0.9999	0.9999	0.01271	0.9999	0.9999	0.0000004
	0.2	0.9999	0.9999	0.00475	0.9999	0.9999	0.0000008
	0.3	0.9999	0.9999	0.01315	0.9999	0.9999	0.0000006
	0.4	0.9999	0.9999	0.2095	0.9999	0.9999	0.0000004
	0.5	0.9999	0.9999	0.1207	0.9999	0.9999	0.0000003

When the same evaluation was conducted for the testing data, it was clear that the model struggled to comprehend the unfamiliar structures (Table 10). The correlation coefficients and MAE scores across all the generated models for the C_i values were poor. MIP and NIP alike had sub 0.5 correlation coefficients, demonstrating that the models had little to no trend, and the error within these was high, as expressed by the MAE score. These values were worse for the NIP with the correlation coefficients being close to zero and no correlation whatsoever. Due to the non-specific nature of the binding towards the NIP this maybe expected, but the statistical data suggest that the model struggled to associate the test structures to any trends that were visible in the training NIP data. Overall, when comparing the trainings statistics to the testing it could be stipulated that 2D structure alone is not enough information to associate with S_b and C_f for the MIP/NIP, with models requiring more information to generate accurate predictions for the values. The limited data size may be another reason for the poor testing results, though the trained data analysis is promising and would initially indicate that the model should perform better than observed.

Table 10. Analysis of the error and correlation of the test datasets at a given C_i value for the ChemCeption model with SMILESToImage featurizer.

	C_i	S_b			C_f		
		R^2	Pearson R^2	MAE Score	R^2	Pearson R^2	MAE Score
MIP	0.1	0.4443	0.0672	4.602	0.0348	0.2888	68.29
	0.2	0.3278	0.2611	11.66	0.0930	0.1238	0.042
	0.3	0.2557	0.5855	13.65	0.3238	0.3816	0.045
	0.4	0.0997	0.00017	22.18953	0.3002	0.3052	0.070
	0.5	0.2529	0.5731	16.45	0.3485	0.3057	0.096
NIP	0.1	0.1178	0.11995	4.9251	0.0350	0.1584	0.025
	0.2	0.1083	0.3891	9.0414	0.5705	0.7338	0.027
	0.3	0.008453	0.00001284	19.4787	0.6429	0.6740	0.030
	0.4	0.01188	0.03938	26.95	0.4790	0.4942	0.059
	0.5	0.07671	0.01253	29.3362	0.6463	0.6872	0.054

When plotting the simulated binding isotherms of each of the tested compounds against that of the experimental data, it is clear that the ChemCeption approach performed poorly (Figure 9). The approach struggled to simulate binding data for all three test molecules with there being little accuracy in the values generated and the trends within the data. This is true for the generation of the datapoints for both MIP and NIP alike, with the model failing to predict non-specific and specific interactions alike. As mentioned previously, this may be an effect of solely analyzing the 2D structures of the molecules and essentially doing an image comparison. The limited amount of data this produces is clearly insufficient to extract complex binding affinity data and to associate such data with molecules in a meaningful manner.

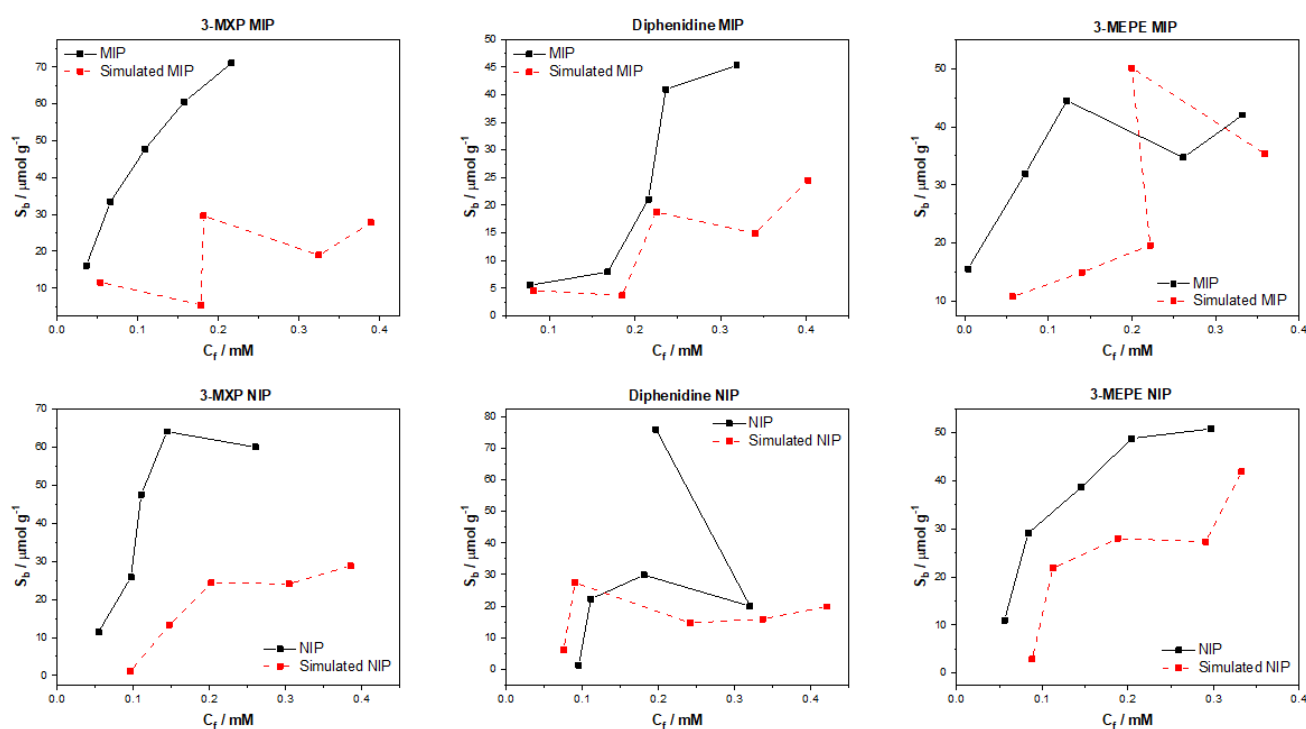


Figure 9. ChemCeption model with SMILEStoImage featurizer simulated and experimental binding isotherms for MIP and NIP in regard to the binding of 3-MXP, diphenidine, and 3-MEPE.

5. Limitations and Considerations

5.1. Optimization

For the optimization of each model, a set of hyperparameters were selected, and these values were based on values that had shown promise in the literature [48]. This being said, there is no real way of knowing if the optimized parameters selected are the optimum for the selected model due to the bias in the range of parameters analyzed. An alternative approach would be to randomly search hyperparameters, building after each iteration towards the more promising set of values randomly tested. A random search itself has certain limitations (upper and lower search limits), though these limitations may prove easier to overcome than the limitations of the current optimization process adopted in this research.

5.2. Computational Limitations

Most models took anywhere between 10–15 min to train and moments to predict values for the testing data, though some of the more sophisticated models/featurizers (DAG and weave) took much longer to train (up to an hour). Therefore, if more sophisticated/complex models were to be used in the future, more time would be required to train the models. The time intensive nature of the computation would also be molecule-specific, with larger molecules taking longer to compute than smaller molecules due to their increased number of molecular features. In the grand scheme of things, this is a small price to pay, though if multiple models are built on top of each other, the calculation time may soon get excessive and slow down the initial rapidness of the technique. A trade-off between model accuracy and time spent training the NN model must be made to ensure the efficiency of the process.

5.3. Limited Dataset

Deep learning is most commonly done on large datasets (hundreds to thousands of datapoints), where trends can easily be established by the selected trained architecture [49]. These large datasets are not currently available for MIP binding data, meaning that the

models trained in the research were using minimal data. This could therefore affect the predictive capabilities of the models, reflecting poor results. This limitation would affect the hyperparameter optimization, making optimum values harder to determine as there are fewer datapoints to build relationships between. A remedy for this could lie in the horizontal expansion of the data, meaning that rather than gathering data on more molecules, the data for the current molecule is expanded. Further physical/quantum properties of the current molecule could be considered, building a more in-depth model between the current dataset molecules and what enables them to interact with a MIP in a specific manner. This said, smaller datasets are also notorious for data over fitting, resulting in a model that has very weak generalization and under performs in comparison to models trained with large datasets. The statistical analysis of the trained and tested models would reflect this flaw, with the trained models showing excellent statistical data and the tested models demonstrating poorer values.

5.4. Selection of Algorithms/Models

The algorithms and approaches selected for the analysis of the MIP/NIP data and the resulting predictions are just a handful of possible approaches out there. These methods represent some of the most common approaches that could be selected, although it should be made clear that there are a lot of other approaches that could be selected for the prediction of the binding isotherms. The selection of simpler machine learning methods could be employed if the objectives of the research were altered slightly. Simpler methods tend to be used in a classification approach (logistic regression, support vector learning), producing binary yes or no predictors, with the possibility of providing probability outcomes. If the research was orientated towards whether a molecule would bind or not bind to a MIP, then it would be possible to use other methods to conduct this research with a higher degree of accuracy/reliability.

6. Conclusions

After testing the multiple algorithms for their potential in predicting MIP/NIP binding affinities, it can be concluded that there is great potential in these approaches for the simulation of MIP data. The most promising of these algorithms include the multitask regression and graph convolution models, with the simulated MIP data proving to be the most accurate representations of the experimental data discussed. The DAG, ChemCep-tion model and graph convolution with weave featurizer proved to have a harder time predicting the binding affinities of molecules, though DAG with modification could have the potential to predict the desired values better. The graph convolution and multitask regressor approaches could predict the trends in the experimental data, producing predicted values that were a fair representation of the testing data. The statistical analysis of all trained and tested models reveals that the models perform well when training but are weak when being tested. As previously mentioned, this is due to the limited dataset and the poor generalization of the models towards different molecules/data. Overall, though the models struggled with over or under predicting the S_b and C_f values, suggesting that additional chemical properties or a combination of algorithms might lead to a more rounded model for the prediction of these values and expansion of the datasets toward other molecules would be fruitful.

The data analyzed and simulated using these approaches were derived from MIP/NIPs synthesized by monolithic bulk polymerization. It should be mentioned that this approach produces relatively heterogeneous MIP/NIP samples with particle sizes and binding cavities randomly distributed throughout the material. The heterogeneous nature is followed into the generated binding isotherms with some of the experimental isotherms proving to have poor trends, and thus making them harder to learn via computational methods. In the future, a more homogenous MIP synthesis could be selected that would therefore lend itself to more homogenous MIP data that could prove easier for machine learning algorithms to learn and predict. Other factors such as MIP composition (monomers, crosslinkers,

and porogens used) and MIP porosity could be factored into the models, building upon reliability and the precision of the approach.

The main take away from the research is that these algorithms can be applied to real world MIP/NIP binding data and can predict affinities that resemble experimental values. This opens up the possibility of utilizing this approach in real-world scenarios, with the manuscript highlighting that it is possible to computationally predict the binding of molecules to MIPs. This is the first instance of this to the authors' knowledge and highlights the potential of machine learning algorithms in this field. The machine learning experiments presented have been conducted by scientists with limited knowledge in the field of machine learning, and have arguably produced results that warrant further investigation into the use of these algorithms in a deep context. If this knowledge was applied by users who have more experience with these models, then we believe that an algorithm and/or approach could easily be identified that could help predict the binding of molecules to specific MIP/NIP compositional chemistries. Thus, this basic research has the potential to open the door for rapid computational simulations of molecular binding affinities and could be used in the rapid analysis of MIPs, reducing material costs, analysis time, and chemical wastage.

Supplementary Materials: The following are available online at <https://www.mdpi.com/article/10.3390/computation9100103/s1>, Table S1: MIP training data. Table S2: NIP training data. Table S3: MIP testing data. Table S4: NIP testing data.

Author Contributions: Conceptualization, J.W.L. and H.D.; methodology, H.I. and J.W.L.; software, H.I. and M.K.K.; validation, J.W.L., K.E. and B.v.G.; formal analysis, M.C. and H.I.; investigation, H.I. and M.K.K.; resources, B.v.G.; data curation, J.W.L., H.I. and M.K.K.; writing—original draft preparation, J.W.L.; writing—review and editing, J.W.L., K.E. and M.C.; visualization, J.W.L.; supervision, H.D., K.E. and T.J.C.; project administration, T.J.C. and H.D.; funding acquisition, B.v.G., K.E. and H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the province of Limburg (The Netherlands), grant “Limburg Meet”, in collaboration with SABER Print.

Data Availability Statement: The data is contained in the article, supplementary information and can also be requested from the corresponding author.

Acknowledgments: The authors are grateful to the DeepChem developer team, especially Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu, for establishing and maintaining the DeepChem infrastructure. The preliminary modelling conducted by Cyrille Sébert and Marc Astner was also greatly appreciated.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. List of Optimized Hyperparameters for the Models

DAG model:

- max_atoms = 50
- n_atom_feat = 50, 75, 100
- n_graph_feat = 30, 40, 50
- n_outputs = 15, 30, 45
- layer_sizes = [100], [150], [200]
- layer_sizes_gather = [100], [150], [200]
- mode = regression
- n_classes = 2

RobustMultitaskRegressor:

- Layer structure: three layers, 1000 nodes each
- Learning rate: 0.0001
- Dropout rate: 0.2
- Featurizer: RDKit descriptors (size = 208)

GraphConvolution:

- Hyperparameters changed from multitask to single task
- Mode: regression
- Number of tasks: 2
- Parameters: S_b , C_f
- Graph convolutional layers: ([256, 256]; [128, 128, 128]; [256, 256])
- Dense layer size: 128
- Dropout rate: (0.001; 0.00001; 0.01)
- Number of atom features: (125; 125; 100)

GraphConvolution (weave featurizer):

- $n_{\text{weave}} = 1, 2$
- Mode: regression
- Number of tasks: 2
- Parameters: S_b , C_f
- Graph convolutional layers: ([256, 256]; [128, 128, 128]; [256, 256])
- Dense layer size: 128
- Dropout rate: (0.001; 0.00001; 0.01)
- Number of atom features: (125; 125; 100)

Chemception:

- Hyperparameters changed from multitask to single task
- Mode: regression
- Augment: TRUE
- Number of tasks: 2
- Parameters: S_b , C_f
- Number of base filters: (32; 32; 16)
- Inception blocks: (A,B,C:3; A,B,C:3; A,B,C:9)

References

1. Nayak, S.; Blumenfeld, N.R.; Laksanasopin, T.; Sia, S.K. Point-of-Care Diagnostics: Recent Developments in a Connected Age. *Anal. Chem.* **2017**, *89*, 102–123. [\[CrossRef\]](#) [\[PubMed\]](#)
2. St John, A.; Price, C.P. Existing and Emerging Technologies for Point-of-Care Testing. *Clin. Biochem. Rev.* **2014**, *35*, 155–167. [\[PubMed\]](#)
3. Kaur, H.; Shorie, M. Nanomaterial based aptasensors for clinical and environmental diagnostic applications. *Nanoscale Adv.* **2019**, *1*, 2123–2138. [\[CrossRef\]](#)
4. Belbruno, J. Molecularly imprinted polymers. *Chem. Rev.* **2019**, *119*, 95–119. [\[CrossRef\]](#)
5. Sharma, P.S.; Iskierko, Z.; Pietrzyk-Le, A.; D'Souza, F.; Kutner, W. Bioinspired intelligent molecularly imprinted polymers for chemosensing: A mini review. *Electrochem. Commun.* **2015**, *50*, 81–87. [\[CrossRef\]](#)
6. Lowdon, J.W.; Diliën, H.; Singla, P.; Peeters, M.; Cleij, T.J.; van Grinsven, B.; Eersels, K. MIPs for Commercial Application in Low-cost Sensors and Assays—An Overview of the Current Status Quo. *Sens. Actuators B Chem.* **2020**, *325*, 128973. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Poma, A.; Guerreiro, A.; Whitcombe, M.J.; Piletska, E.V.; Turner, A.P.F.; Piletsky, S.A. Solid-phase synthesis of molecularly imprinted polymer nanoparticles with a reusable template—"Plastic Antibodies". *Adv. Funct. Mater.* **2013**, *23*, 2821–2827. [\[CrossRef\]](#)
8. Ambrosini, S.; Beyazit, S.; Haupt, K.; Bui, B.T.S. Solid-phase synthesis of molecularly imprinted nanoparticles for protein recognition. *Chem. Commun.* **2013**, *49*, 6746–6748. [\[CrossRef\]](#)
9. Dorko, Z.; Nagy-Szakolczai, A.; Toth, B.; Horvai, G. The selectivity of polymers imprinted with amines. *Molecules* **2018**, *23*, 1298. [\[CrossRef\]](#)
10. Alenazi, N.A.; Manthorpe, J.M.; Lai, E.P.C. Selectivity enhancement in molecularly imprinted polymers for binding of bisphenol A. *Sensors* **2016**, *16*, 1697. [\[CrossRef\]](#)
11. Feng, F.; Lai, L.; Pei, J. Computational chemical synthesis analysis and pathway design. *Front. Chem.* **2018**, *6*, 199. [\[CrossRef\]](#)
12. Cavasotto, C.N.; Aucar, M.G.; Adler, N.S. Computational chemistry in drug lead discovery and design. *J. Quantum. Chem.* **2019**, *199*, 25678. [\[CrossRef\]](#)
13. Siddique, N.; Adeli, H. Nature inspired chemical reaction optimisation algorithms. *Cogn. Comput.* **2017**, *9*, 411–422. [\[CrossRef\]](#)
14. Mamo, S.K.; Elie, M.; Baron, M.G.; Gonzalez-Rodriguez, J. Computationally designed perhenate ion imprinted polymers for selective trapping of rhenium ions. *ACS Appl. Polym. Mater.* **2020**, *2*, 3135–3147. [\[CrossRef\]](#)

15. Woznica, M.; Sobiech, M.; Palka, N.; Lulinski, P. Monitoring the role of enantiomers in the surface modification and adsorption process of polymers imprinted by chiral molecules: Theory and practice. *J. Mater. Sci.* **2020**, *55*, 10626–10642. [[CrossRef](#)]
16. Cowen, T.; Busato, M.; Karim, K.; Piletsky, S.A. Insilico synthesis of synthetic receptors: A polymerization algorithm. *Macromolecules* **2016**, *37*, 2011–2016.
17. Sobiech, M.; Zolek, T.; Lulinski, P.; Maciejewske, D. A computational exploration of imprinted polymer affinity based on coriconazole metabolites. *Analyst* **2014**, *139*, 1779–1788. [[CrossRef](#)] [[PubMed](#)]
18. Maryasin, B.; Marquetand, P.; Maulide, N. Machine learning for organic synthesis: Are robots replacing chemists? *Angew. Chem.* **2018**, *57*, 6978–6980. [[CrossRef](#)]
19. Balaraman, M.; Soundar, D. An artificial neural network analysis of porcine pancreas lipase catalyzed esterification of anthranilic acid with methanol. *Proc. Biochem.* **2005**, *40*, 3372–3376.
20. Betiku, E.; Taiwo, A.E. Modeling and optimization of bioethanol production from breadfruit starch hydrolysate vis-à-vis response surface methodology and artificial neural network. *Renew. Energy* **2015**, *74*, 87–94. [[CrossRef](#)]
21. Mater, A.C.; Coote, M.L. Deep learning in chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545–2559. [[CrossRef](#)] [[PubMed](#)]
22. Tkatchanko, A. Machine learning for chemical discovery. *Nat. Commun.* **2020**, *11*, 4125. [[CrossRef](#)] [[PubMed](#)]
23. Mohapatra, S.; Dandapat, S.J.; Thatoi, H. Physicochemical characterization, modelling and optimization of ultrasono-assisted acid pretreatment of two pennisetum sp. Using Taguchi and artificial neural networking for enhanced delignification. *J. Environ. Manag.* **2017**, *187*, 537–549. [[CrossRef](#)]
24. Ramsundar, B.; Kearnes, S.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively Multitask Networks for Drug Discovery. *arXiv* **2015**, arXiv:1502.02072.
25. Jiao, Z.; Hu, P.; Xu, H.; Wang, Q. Machine learning and Deep learning in chemical health and safety: A systematic review of techniques and applications. *ACS Chem. Health Saf.* **2020**, *27*, 316–334. [[CrossRef](#)]
26. Sun, M.; Zhao, S.; Gilvary, C.; Elemento, O.; Zhou, J.; Wang, F. Graph convolutional networks for computational drug development and discovery. *Brief. Bioinform.* **2020**, *21*, 919–935. [[CrossRef](#)]
27. Gonzalez-Diaz, H.; Perez-castillo, Y.; Podda, G.; Uriarte, E. Computational chemistry comparison of stable/nonstable protein mutants classification models based on 3D and topological indices. *J. Comp. Chem.* **2007**, *28*, 1990–1995. [[CrossRef](#)]
28. Feinberd, E.N.; Joshi, E.; Pande, V.S.; Cheng, A.C. Improvement in ADMET prediction with multitask deep featurization. *J. Med. Chem.* **2020**, *63*, 8835–8848. [[CrossRef](#)]
29. Jeon, W.; Kim, D. FP2VEC: A new molecular featurizer for learning molecular properties. *Bioinformatics* **2019**, *35*, 4979–4985. [[CrossRef](#)]
30. Toropov, A.A.; Toropova, A.P.; Benfenati, E.; Leszczynska, D.; Leszczynski, J. SMILES-based optimal descriptors: QSAR analysis of fullerene-based HIV-1 PR inhibitors by means of balance of correlations. *J. Comp. Chem.* **2009**, *31*, 381–392. [[CrossRef](#)]
31. Baskin, I.I.; Winkler, D.; Tetko, I.V. A renaissance of neural networks in drug discovery. *Expert Opin. Drug Discov.* **2016**, *11*, 785–795. [[CrossRef](#)]
32. Zhang, J.; Zhao, J.; Lin, H.; Tan, Y.; Cheng, J. High-speed chemical imaging by dense-net learning of femtosecond stimulated raman scattering. *J. Phys. Chem. Lett.* **2020**, *11*, 8573–8578. [[CrossRef](#)]
33. Okuwaki, J.; Ito, M.; Mochizuki, Y. Development of educational scratch program for machine learning for students in chemistry course. *J. Comp. Chem.* **2020**, *18*, 126–128.
34. Hutchinson, S.T.; Kobayashi, R. Solvent-specific featurization for predicting free energies of solvation through machine learning. *J. Chem. Inf. Mod.* **2019**, *59*, 1338–1346. [[CrossRef](#)] [[PubMed](#)]
35. Janczura, M.; Lulinski, P.; Sobiech, M. Imprinting technology for effective sorbent fabrication: Current state-of-the-art and future prospects. *Materials* **2021**, *14*, 1850. [[CrossRef](#)] [[PubMed](#)]
36. Suryana, S.; Rosandi, R.; Hasanah, A.N. An update on molecularly imprinted polymer design through a computational approach to produce molecular recognition material with enhanced analytical performance. *Molecules* **2021**, *26*, 1891. [[CrossRef](#)] [[PubMed](#)]
37. Zapata, F.; Matey, J.M.; Montalvo, G.; Garcia-Ruiz, C. Chemical classification of new psychoactive substances (NPS). *Microchem. J.* **2021**, *163*, 105877. [[CrossRef](#)]
38. Castaing-Cordier, T.; Ldroue, V.; Besacier, F.; Bulete, A.; Jacquemin, D.; Giraudeau, P.; Farjon, J. High-field and benchtop NMR spectroscopy for the characterization of new psychoactive substances. *Forensic Sci. Int.* **2021**, *321*, 110718. [[CrossRef](#)] [[PubMed](#)]
39. Valli, A.; Lonati, D.; Locatelli, C.A.; Buscaglia, E.; Tuccio, M.D.; Papa, P. Analytically diagnosed intoxication by 2-methoxyphenidine and flubromazepam mimicking an ischemic cerebral disease. *Clin. Toxicol.* **2017**, *55*, 611–612. [[CrossRef](#)]
40. Luethi, D.; Hoener, M.C.; Liechti, M.E. Effects of the new psychoactive substances diclofensine, diphenidine, and methoxyphenidine on monoaminergic systems. *Eur. J. Pharmacol.* **2018**, *819*, 242–247. [[CrossRef](#)]
41. Lowdon, J.W.; Eersels, E.; Rogosic, R.; Heidt, B.; Dillen, H.; Steen Redeker, E.; Peeters, M.; van Grinsven, B.; Cleij, T.J. Substrate displacement colorimetry for the detection of diarylethylamines. *Sens. Actuators B Chem.* **2019**, *282*, 137–144. [[CrossRef](#)]
42. Lowdon, J.W.; Alkirk, S.M.O.; Mewis, R.E.; Fulton, D.; Banks, C.E.; Sutcliffe, O.B.; Peeters, M. Engineering molecularly imprinted polymers (MIPs) for the selective extraction and quantification of the novel psychoactive substance (NPS) methoxyphenidine and its regioisomers. *Analyst* **2018**, *9*, 2002–2007. [[CrossRef](#)] [[PubMed](#)]
43. Lusci, A.; Pollastri, G.; Baldi, P. Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. *J. Chem. Inf. Model.* **2013**, *53*, 1565–1575. [[CrossRef](#)] [[PubMed](#)]

-
44. Son, J.; Kim, D. Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. *PLoS ONE* **2021**, *16*, e0249404. [[CrossRef](#)] [[PubMed](#)]
 45. Tan, Z.; Li, Y.; Shi, W.; Yang, S. A multitask approach to learn molecular properties. *J. Chem. Inf. Model.* **2021**, *61*, 3824–3834. [[CrossRef](#)] [[PubMed](#)]
 46. Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: Moving beyond fingerprints. *J. Comp. Mol. Des.* **2016**, *30*, 595–608. [[CrossRef](#)] [[PubMed](#)]
 47. Coley, C.W.; Green, W.H.; Jensen, K.F. RDChiral: An RDKit wrapper for handling stereochemistry in retrosynthetic template extraction and application. *J. Chem. Inf. Model.* **2019**, *59*, 2529–2537. [[CrossRef](#)]
 48. Yoo, Y. Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches. *Knowl. Based Syst.* **2019**, *178*, 74–83. [[CrossRef](#)]
 49. D'souza, R.N.; Huang, P.; Yeh, F. Structural analysis and optimization of convolutional neural networks with a small sample size. *Sci. Rep.* **2020**, *10*, 834. [[CrossRef](#)]