Neural Computing and Applications
https://doi.org/10.1007/s00521-021-05961-4

ORIGINAL ARTICLE

# Prediction of chemical compounds properties using a deep learning model

Mykola Galushka[1] · Chris Swain[2] · Fiona Browne[3] · Maurice D. Mulvenna[4] · Raymond Bond[4] · Darren Gray[5]

Received: 8 September 2020 / Accepted: 25 March 2021


## Abstract

The discovery of new medications in a cost-effective manner has become the top priority for many pharmaceutical companies. Despite decades of innovation, many of their processes arguably remain relatively inefficient. One such process is the prediction of biological activity. This paper describes a new deep learning model, capable of conducting a preliminary screening of chemical compounds in-silico. The model has been constructed using a variation autoencoder to generate chemical compound fingerprints, which have been used to create a regression model to predict their LogD property and a classification model to predict binding in selected assays from the ChEMBL dataset. The conducted experiments demonstrate accurate prediction of the properties of chemical compounds only using structural definitions and also provide several opportunities to improve upon this model in the future.

**Keywords** Machine learning · Deep neural networks · chemical compounds Properties

**Abbreviation**

| | |
|---|---|
| ASCII | American Standard Code for Information Interchange |
| ChEMBL | Chemogenomic European Molecular Biology Laboratory |
| DAG | Directed Acyclic Graph |
| EBI | European Bioinformatics Institute |
| EC50 | Half maximal effective concentration |
| ECFP | Extended-Connectivity Fingerprints |
| EMBL | European Molecular Biology Laboratory |
| GC | Graph Convolutional |
| HTC | High-Throughput Screening |
| IC50 | half maximal inhibitory concentration |
| IRV | Influence Relevance Voting |
| Kd | Dissociation constant |
| Ki | Inhibition constant |
| kNN | k-Nearest Neighbour |
| KRR | Kernel Ridge Regression |
| LR | Logistic Regression |
| MLP | Multilayer Perceptron |
| MNN | Multitask Neural Network |
| MPNN | Message Passing Neural Network |
| MUV | Maximum Unbiased Validation |
| pH | potential of hydrogen |
| RF | Random Forests |
| ROC-AUC | Receiver Operating Characteristic Area Under Curve |
| SMILES | Simplified Molecular-Input Line-Entry System |

✉ Maurice D. Mulvenna
md.mulvenna@ulster.ac.uk

Mykola Galushka
mm.galushka@auromind.org

Chris Swain
swain@mac.com

Fiona Browne
fiona.browne@datactics.com

Raymond Bond
rb.bond@ulster.ac.uk

Darren Gray
darren.gray@almacgroup.com

[1] AUROMIND Ltd., 126 Eglantine Avenue, Belfast BT9 6EU, UK

[2] Cambridge MedChem Consulting, 8 Mangers Lane, Duxford, Cambs CB22 4RN, UK

[3] Datactics Ltd., One Lanyon Quay, Belfast BT1 3LG, UK

[4] School of Computing, Ulster University, Jordanstown BT37 0QB, UK

[5] Almac Sciences Ltd., 20 Seagoe Industrial Estate, Craigavon BT63 5QD, UK

Published online: 04 June 2021

XGB eXtreme Gradient Boosting

## 1 Introduction

Deep learning [24] has been successfully applied in a number of problem domains from natural language processing [39], medical imaging analysis [32] to finance [26]. Deep learning architectures are also successfully used for many predictive tasks in chemistry and biology domains [55]. One application of deep learning in the chemistry domain is to predict important properties of chemical compounds. It allows for the assessment of chemical compounds before committing to an expensive synthesis process [7, 8, 20, 43].

Deep learning theory is based on deep neural networks (DNN) which consist of many layers. Each layer is comprised of a number of neurons. Higher levels of the DNN represent more complex concepts. To improve network performance, layers are often implemented using different methodologies. The overall topology of a neural network is selected based on the problem to be solved and often is tuned during an experimental phase.

In this paper, the DNN learns to represent compounds by their chemical descriptors. This opens a wide range of opportunities to build sophisticated machine learning applications for predicting different properties of chemical compounds.

This research study investigates how the pretrained *autoencoder* can be used for building *classification* and *regression* models for predicting the *LogD* property and target *binding* of chemical compounds using data obtained from different sources.

The remainder of the paper is outlined as follows. Section 2 provides background information which includes a description of predicting properties of chemical compounds and other machine learning models used for their prediction. Section 3 introduces the mathematics behind the developed *autoencoder* together with visualizations of the learning mechanisms. Section 4 provides detail on the data sets used and experiments carried out, while Sect. 5 describes the obtained results, followed by discussion and conclusions in Sect. 6.

## 2 Background

### 2.1 Predicting properties

#### 2.1.1 LogD

Lipophilicity is possibly one of the most important physicochemical properties of a potential drug. It plays a role in solubility, absorption, membrane penetration, plasma protein binding, distribution, CNS penetration and partitioning into other tissues or organs such as the liver and has an impact on the routes of clearance. It is important in ligand recognition, not only to the target protein but also CYP450 interactions, HERG binding, and PXR mediated enzyme induction. Most drugs entering a market are designed for oral administration. The absorption of drugs can either be via passive diffusion across membranes or via carrier mediated transport. Carrier mediated transport is energy dependent and requires a specific transporter protein. In contrast passive diffusion does not require the presence of a specific carrier transporter protein and is less structure specific than carrier-mediated transport; there is a general dependence on lipophilicity for structurally diverse compounds. However, the relationship with LogD is non-linear with an optimum of LogD 1-2.

Measurement of LogP can be undertaken in a variety of ways, the most common is the shake-flask method, which consists of dissolving some of the solute in question in a volume of octanol and water, shaking for a period of time, then measuring the concentration of the solute in each solvent. This can be time-consuming particularly if there is no quick spectroscopic method to measure the concentration of the molecule in the phases. A faster method of logP determination makes use of high-performance liquid chromatography.

However, the majority of known drugs contain ionizable groups, as shown in Fig. 1, which shows the distribution of small molecule drugs with DrugBank [53] and are likely to be charged at physiological pH and LogP only correctly describes the partition coefficient of neutral (uncharged) molecules. LogD, the distribution constant is a better descriptor of the lipophilicity of a molecule. This can be determined in a similar manner to LogP but instead of using water, the aqueous phase is adjusted to a specific pH using a buffer. LogD is thus pH dependent, hence one must specify the pH at which the logD was measured. Of

particular interest is the logD at pH = 7.4 (the physiological pH of blood serum).

Usually, it is not practical to determine the LogD of every compound made experimentally (and it may be of interest to calculate logD prior to synthesis) and so calculated results are used.

#### 2.1.2 Binding

The majority of drug-like small molecules are specifically designed to bind to protein targets involved in disease related pathways. The activity of molecules in a biological assay may be captured by a variety of different measures IC50, EC50, Ki, % inhib, etc., but most are a measure of a binding event in some manner. The biological results varies considerably in quality from single point high-throughput screening (HTS) data [15] to full dose response curves. Much of these data are captured in ChEMBL, a database of bioactive drug-like small molecules and abstracted bioactivities.

### 2.2 Models for comparison

The developed DNN model provides an accurate prediction of *LogD* and *binding* properties. To gauge its performance, a total of ten machine learning (ML) techniques were used. Taking into consideration a large variety of different experimental setups, model implementations and evaluation metrics, authors tried to summarize the results from a number of sources and provide a 'single' performance metric to compare each of the techniques.[1]

- Logistic regression (LR) [41] is a very straightforward and popular classification model, which has a long and successful history of been used in many ML applications and statistical modelling. It uses a logistic function for weighting a linear combination of input parameters. LR models have been used in a large number of publications involving chemical data types.
- Kernel ridge regression (KRR) [57] uses a modified approach to find a regression function by adding a bias, which causes a drop in variance. In other words a better prediction can be achieved by considering a slightly less fit model during the training process.
- Another very popular and well recognized model is random forests (RF) [56]. It can be applied for both classification and regression problems and uses an ensemble of decision trees, which are trained on different subsets of the original data.

[1] This assessment takes into the account, all aspects of carried out experiments and selects results from those, which correlate with tests performed in the current study.

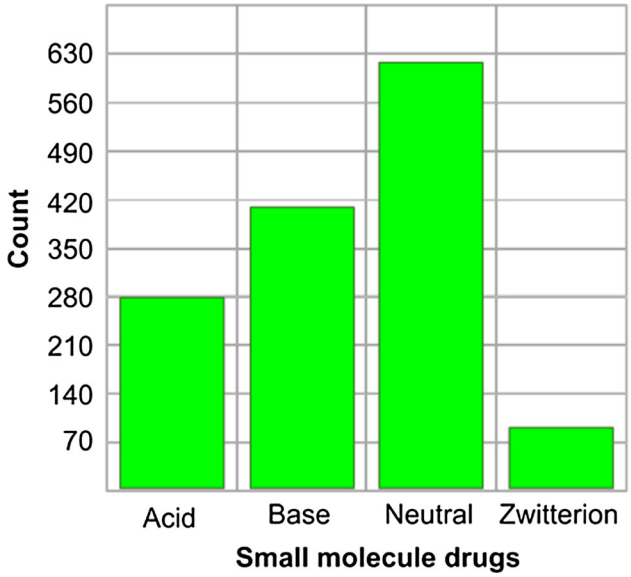


**Fig. 1** Distribution of small molecule drugs with DrugBank

- Extreme gradient boosting (XGB) [14] tree is similar to RF and is an ensemble method. During each training step it constructs a new tree model, which in combination with previous models, minimizes the overall prediction error. It is a very popular approach in the ML community, and has been successfully applied to many problems, consistently providing accurate results.
- Multitask neural network (MNN) [49] is a special modified neural network architecture for solving simultaneously multiple problems. There are a number of different designs of these neural networks. However, they are all constructed based on the principle that some fully-connected layers are shared between different tasks. In this way training processes for solving one task can influence other tasks and vice versa.
- Graph convolutional (GC) [18] models are designed to utilize molecules which can be transformed into undirected graphs where atoms are represented as nodes and bonds as edges respectively. A convolution is applied to expand a feature space, by creating multiple filters representing graph substructures. The aggregation of these substructures is performed via multiple convolution layers.
- Message passing neural network (MPNN) [23] model is based on mathematical framework, which generalizes a number of graph-based neural network designs. It performs computation in two phases: message passing phase (constantly updates a hidden state) and read-out phase (uses the final hidden state, used for making a prediction).
- Directed acyclic graph (DAG) [37] is another popular approach creating *classification* and *regression* models. It is based on a chemical compounds graph structure.

The previously reviewed graph-based models are using undirected graphs (where molecular bonds between atoms naturally do not have directions). However, the latest DAG design synthetically introduces an additional directional feature for the graph-based molecular representation. It identifies a central atom and creates a directional-structure of chemical compound from this central point. Generating the additional features may provide an improvement in prediction accuracy in comparison to other graph-based models.

- WEAVE [31] supports graph-based model that utilizes both properties of chemical compounds: nodes (for atoms) and edges (for bonds). The constructed features matrix is processed by convolution-like filters. Similar to image convolution neural network, WEAVE provides more informative representations of chemical compound structures.
- Influence relevance voting (IRV) [51] is not the most common approach, but it has certain important advantages against other ML models. Its prediction can be easily interpreted in a similar fashion to k-nearest neighbours (kNN). IRV tries to identify k-nearest neighbours using a neural network to compute a more complex similarity function.

The majority of these models rely on two key factors: availability of large volumes of training data and knowledge about the physical structure of chemical compounds (which can be used for converting them to graphs or fingerprints). Despite large collections of stock chemical compounds, an offering by many research and commercial entities, the number of compounds in individual assays investigating a specific problem, is relatively small. It may significantly impact the performance of many ML models. Besides, if a model relies on the physical structure of chemical compounds the training process can become very computationally expensive.

This study investigates transfer learning using the *variational autoencoder*. It reduces reliance on a large volume of training data from the specific assays.[2] This model also does not require any external knowledge about the physical structure of chemical compounds. All knowledge required for making accurate predictions derive from SMILES representation of chemical compounds.

[2] The variational auto-encoder can be trained to generate chemical compounds fingerprints using a large database of compounds such as ChEMBL. Then, this model can use a small portion of training data (from assays), to predict specific chemical compounds properties of binding characteristics.

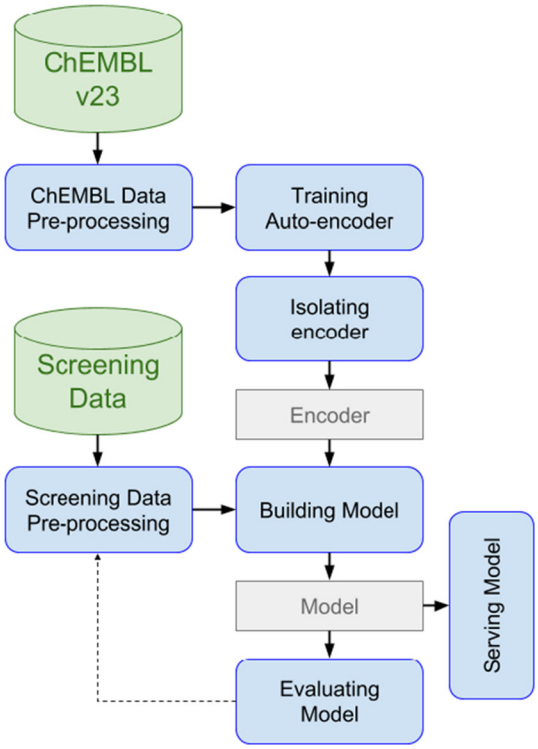


**Fig. 2** A summary of the proposed approach where ChEMBL data are used to train the *autoencoder* and screening data are used to build and evaluate a model for predicting desirable properties of chemical compounds

## 3 Methodology

The proposed methodology uses a pretrained *autoencoder* to build *classification* and *regression* models for predicting properties of chemical compounds. A high-level overview of the proposed approach is presented in Fig. 2.

The majority of the proposed workflow stays the same for predicting LogD and binding properties. It starts with selecting a collection of chemical compounds from the ChEMBLv23 database [11]. These compounds are used for training *variational autoencoder*. Then, a specially designed process isolates encoder layers of the *variational autoencoder*. These layers are combined with an additional sub-network for performing classification or regression tasks (depending on the problem being solved). The constructed regression or classification neural networks are trained using screening data derived from HTS. It is important to mention that the encoder layers remain frozen throughout this training. The obtained model is used for evaluation, and if the assessment is successful, the model can be deployed in a production environment.

The rest of the methodology section is split into three parts. Section 3.1 describes a *variational autoencoder*. Section 3.2 focuses on *classification* and *regression* models for predicting desirable properties of chemical compounds.

Section 3.3 explains a learning process using a simple and intuitive example.

### 3.1 Variational autoencoder

An *autoencoder* is a neural network which can be used to address representational learning problems [30]. It learns to reconstruct the original input using an informational bottleneck. Such neural networks have been successfully applied in various domains such as noise filtering in audio-video content, translation, and compression. The proposed approach focuses on the *autoencoder* for reconstruction of chemical compounds. It takes simplified molecular-input line-entry system (SMILES) [52] and tries to reproduce these SMILES using latent space. SMILES are ASCII strings for describing the structure of chemical species. They can be imported by most molecule editors and converted into two-dimensional or three-dimensional structures of the molecules. For example **Aspirin** $C_9H_8O_4$ is represented by the following SMILES: `CC(=O)OC1=CC=CC=C1C(=O)O`, whose 2D structure is shown in Fig. 3.

A generic *autoencoder* is trained to minimize the reconstruction error $\mathcal{L}$ defined by Eq. 1:

$$\min\{\mathcal{L}(x,\hat{x})+\mathcal{R}\}, \tag{1}$$

where $x$ is the original input, $\hat{x}$ is the reconstructed output and $\mathcal{R}$ is a regularizer. The regularizer penalizes a model using large weights to prevent memorization and overfitting problems [17]. Ideally, a well-trained *autoencoder* should accurately reconstruct an input SMILES $x$ such as already mentioned `CC(=O)OC1=CC=CC=C1C(=O)O`.

In this paper, we are focusing on *variational autoencoder* [22], a special type of *autoencoder*, which uses a probability distribution to reconstruct the original input $x$. Suppose $z$ represents hidden variables (from the latent space) used to reconstruct the original input:

$$p(z|x)=\frac{p(x|z)p(z)}{p(x)}=\frac{p(z,x)}{p(x)}. \tag{2}$$

The computation of the marginal distribution $p(x)$ is very complex, since the following integral is intractable (in majority cases):

$$p(x)=\int p(x|z)p(z)dz. \tag{3}$$

There are two main approaches available to tackle this problem: Monte Carlo [38] and variational inference (used to build *variational autoencoder*) [27]. Let's approximate $p(z|x)$ with another distribution $q(z|x)$, where $q$ can be chosen as a tractable distribution (such as Gaussian) [29]. Then, it is possible to find distribution parameters when $q$ becomes close enough to $p$ by minimizing the Kullback–Leibler divergence [47], which measures an amount of lost information for the chosen approximation:

$$KL(q(z|x)||p(z|x))=-\sum q(z|x)\log\frac{p(z|x)}{q(z|x)}. \tag{4}$$

By replacing $p(z|x)$ in Eq. 4 it is possible to derive to the following equation:

$$KL(q(z|x)||p(z|x))=-\sum q(z|x)\log\frac{p(z,x)}{q(z|x)}+p(x), \tag{5}$$

and express $p(x)$ as:

$$p(x)=KL(q(z|x)||p(z|x))+\sum q(z|x)\log\frac{p(z,x)}{q(z|x)}, \tag{6}$$

where the second term is called a *variational low bound*:

$$\mathcal{L}=\sum q(z|x)log\frac{p(z,x)}{q(z|x)}. \tag{7}$$

This allows us to rewrite Eq. 6 as:

$$p(x)=KL(q(z|x)||p(z|x))+\mathcal{L}, \tag{8}$$

where $p(x)$ in Eq. 8 can be considered as a constant, since $x$ is given (as the original input). The rest of this equation represents a sum of two quantities, where KL-divergence needs to be minimized. The minimization of KL-divergence is effectively a maximization of the variational low bound $\mathcal{L}$ defined by Eq. 7. By substituting $p(z, x)$ in Eq. 7 it is possible to derive to the following equation:

$$\mathcal{L}=\sum q(z|x)\log p(x|z)+\sum q(z)\log\frac{p(z)}{q(z|x)}, \tag{9}$$

where $p(x|z)$ is the expectation with respect to $q(z)$ and can be written as $E_{q(z)}\log p(x|z)$. The second term $-KL(q(z|x)||p(z|x))$ represents the KL-divergence. This allows us to rewrite Eq. 9 as following:

$$\mathcal{L}=E_{q(z)}\log p(x|z)-KL(q(z|x)||p(z|x)). \tag{10}$$

Let's build the *variational autoencoder* based on variational low bound Eq. 9. The observed distribution $q$ is a function mapping $x$ to $z$, which should match an another distribution $p$. The observed distribution $p$ is a function mapping $z$ to $\hat{x}$, where $p$ can be chosen. Both $q$ and $p$ are implemented as neural networks and for the further references are called *encoder* and *decoder* accordingly. A visualization of the *variational autoencoder* model is shown in Fig. 4.

Let's select $p$ to be the Gaussian distribution. This requires us to make the distribution $q$ (in the latent layer) also similar to Gaussian. The cost function can then be expressed as:

$$\min|x-\hat{x}|^2-KL(q(z|x)||N(\mu,\sigma)), \tag{11}$$

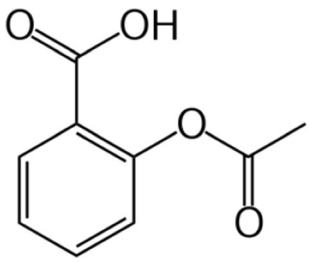


**Fig. 3** Aspirin 2D structure

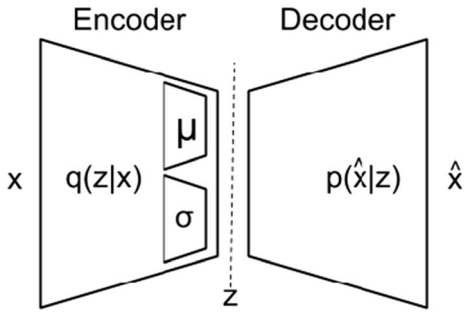


**Fig. 4** A schema of *variational autoencoder* model

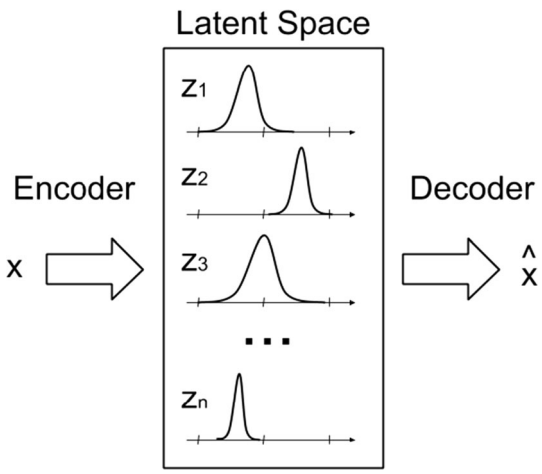


**Fig. 5** The mechanism of the input reconstruction using latent space

where $N$ is a normal distribution defined by two parameters $\mu$-mean and $\sigma$-variance. $|x - \hat{x}|^2$ in 11 has been derived from the definition of the reconstruction error for the Gaussian distribution $p(x|\hat{x}) = e^{-|x-\hat{x}|^2}$.

As shown in Fig. 5, the *encoder* learns to represent the original input $x$ as a set of attributes $z$ in the latent space, where each attribute is defined as the probability distribution (with parameters $\mu$ and $\sigma$). The *decoder* learns to reconstruct $\hat{x}$ close to the original input $x$ using a set of attributes $z$ from the latent space.

A set of attributes $z$ in the latent space represents chemical compounds fingerprints and can be used for building *classification* and *regression* models. A typical

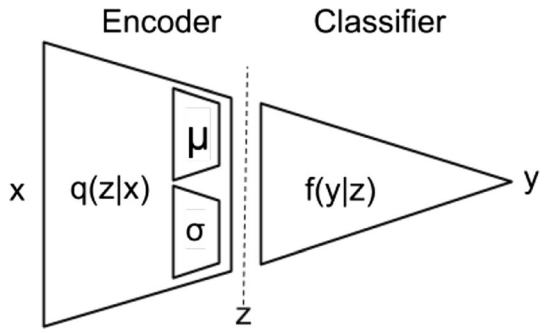


**Fig. 6** A schema of built *classification* and *regression* models based on the *variational autoencoder*

approach for building *classification* and *regression* models is to join together trained *encoder* and problem-dependent prediction layers as shown in Fig. 6.

The training process in such architecture learns some function $f(y|z)$ which predicts $y$ (category value for a classification problem or real value for a regression problem) using chemical compounds fingerprints $z$ generated by *encoder*. In the classical approach the *encoder* neural network $q(z|x)$ only generates $z$ and does not take part in the training process (so it weights remain frozen thorough out all training cycle).[3]

### 3.2 Architecture

The *variational autoencoder* was implemented using a Convolution Neural Network [6] in combination with a few layers for supporting the variational training process. Its neural network topology is presented in Fig. 7.

It consists of two joined neural networks: *encoder* and *decoder*. The *encoder* neural network consists of nine layers. The *Input* layer takes SMILES transformed to the one-hot 150x78 matrix representation (each row represents a SMILES character and column its encoding). A visualization of SMILES transformation is shown in Fig. 8.

This figure demonstrates a process of one-hot encoding for the Aspirin SMILES, introduced earlier. If the input SMILES is less than 150 characters, the original sequence is padded with empty spaces on the right. When the input sequence is adjusted, the transformation process creates a zero-matrix with 150 rows (equals to the maximum number of characters in the input sequence) and 78 column (equals to SMILES vocabulary size). A transformation loop selects each character in the padded sequence and use a dictionary to identify the character code. A position of the selected

[3] The conducted experiments showed that relaxing *encoder* weights may help to improve prediction accuracy and will be discussed in the following sections.

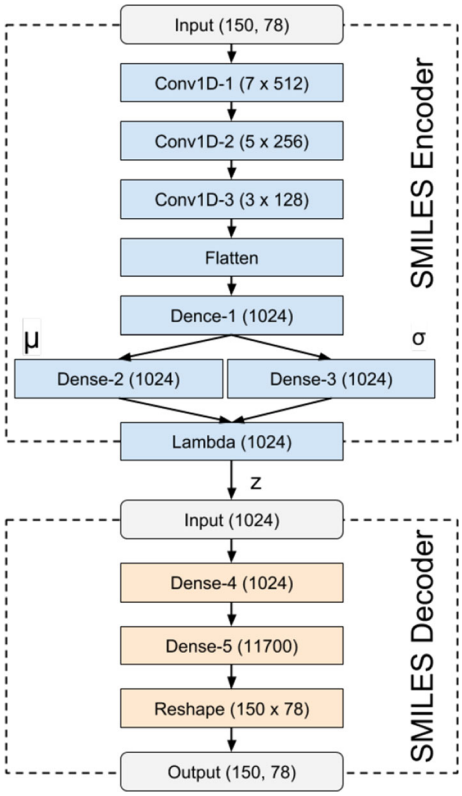


**Fig. 7** A neural network topology of *variational autoencoder*

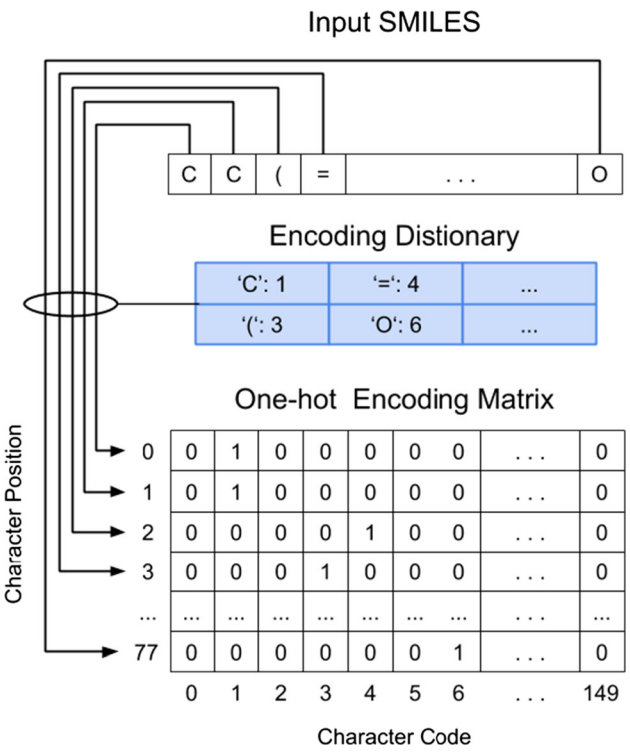


**Fig. 8** SMILES one-hot encoding

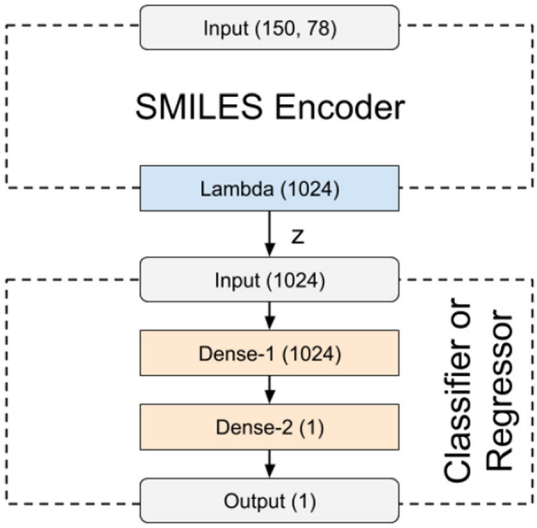


**Fig. 9** A neural network topology of *classification* and *regression* models

character and its code are used to set the according element to 1 in the one-hot matrix.[4]

The input layer is followed by three 1D-convolution layers (*Conv1D-1*, *Conv1D-2* and *Conv1D-3*) with 512, 256, 128 filters and 7, 5, 3 kernel sizes accordingly. These convolution layers perform a very similar role to 2D-convolution layers in image processing. They identify specific patterns of elements (atom and bonds) in chemical compounds and aggregate them in bigger substructures at each consequent layer. A useful insight of how these layers work is presented in the discussion Sect. 6.

The *Flatten* layer vectorizes convolution weights, so they can be processed by the following *Dense-1* layer. This layer has 1024 neurons, which is exactly the same size as an output latent vector $z$. The size of this layer has been defined via a hyper-parameters tuning procedure. *Dense-2* and *Dense-3* layers implement variational learning, which computes $\mu$ and $\sigma$ accordingly. The final *Lambda* layer combines $\mu$ and $\sigma$ into a single latent vector (consisting of 1024 real values). Effectively this latent vector represents a chemical compound fingerprint.

The *decoder* consists of five layers. The input layer receives the latent vector from the *encoder* and passes it via two dense layers: *Dense-4* and *Dense-5*. The *Dense-5* scale-up original dimensionality from 1024 to 11700. This step is needed to transform a 1D into a 2D vector which is performed by the *Reshape* layer. This layer passes a 2D vector straight to the output. The final output of all these layers is 150x78 matrix, which can be reversed to the SMILES sequence.

[4] The reverse process is applied to reconstruct the original sequence from the one-hot matrix representation.

A neural network topology of *classification* and *regression* models constructed based on pretrained encoder is shown in Fig. 9.

The *encoder* in *classification* and *regression* models has exactly the same topology structure as already described in *variational autoencoder*. However, the attached layers are designed to perform classification or regression tasks. The *Dense-1* layer receives chemical compounds fingerprints and passes it to the next layer *Dense-2*. There is no difference in the current neural network topology whether it's applied for a classification or regression problem. The only difference is in an inactivation for the *Dense-2* layer.[5] In case of the regression problem the activation function as *relu* and in case of the classification problem the activation function as *sigmoid*.

Due to the high level of complexity defined above methodology and architecture, it is helpful to review a simple example, which illustrates a prediction process.

### 3.3 Prediction example

An example of neural network classifier built based on an *encoder* is shown in Fig. 10.

This is a high level of visualization intends to demonstrate some key concepts described above. Let us assume we have three types of compounds: red, green and blue. The red and green compounds have a very similar rectangular shape (despite the green compound has slightly rounded edges). The blue compound has the triangle shape. A constructed neural network should identify two classes: rectangle or triangle for the input compound.

According to the *variational autoencoder* methodology the trained *encoder* generates a latent vector representation for each input compound. This latent vector is a 'compact' representation[6] of the original input and in chemistry domain can be also refereed as a chemical compound fingerprint. It is likely that similar inputs will have a similar latent vector representation. It also means that similar compounds should be closely located in the latent space. The latent space is an abstract concept, which can be very useful to visualize a distributing of encoded compounds. As it can be seen from Fig. 10 red and green crosses represent location of 'rectangular' instances in close approximation from each other. The blue cross represents a 'triangular' sample accordingly. The dashed line represents areas of distribution of compounds with similar shapes in the latent space.

MLP neural network can provide an efficient architecture to learn a distribution of compounds in the latent space. Since the latent vector has much smaller size in comparison to the original input, MLP does not require a complex topology. One or two hidden layers can be sufficient to handle prediction of properties of chemical compounds for the majority of cases.

The next section describes a series of experiments for evaluating the proposed solutions.

## 4 Data and experiment design

In order to evaluate the proposed methodology three studies are presented in this paper:

- An evaluation of the *variational autoencoder* for reconstruction of SMILES Sect. 4.1.
- An evaluation of regression model for predicting LogD properties of chemical compounds Sect. 4.2.
- An evaluation of classification model for drug-target predictions Sect. 4.3.

### 4.1 Experimental data and setup for SMILES reconstruction problem

ChEMBL is a chemical database of bio-active molecules with drug-like properties [16]. It is maintained by the European Bioinformatics Institute (EBI) of the European Molecular Biology Laboratory (EMBL), located at the Wellcome Trust Genome Campus in Hinxton, UK. ChEMBL data are widely used by pharmaceutical companies and research organizations around the World for creating screening libraries in drug discovery. ChEMBLv23 (version 23) has been selected for the current study. It includes approximately 1.7M chemical compounds.

The initial study [21] already showed an accurate SMILES reconstruction using a *variational autoencoder* neural network. This work is focusing on an optimization of a training process. To train a neural network based on all chemical compounds containing in ChEMBL already requires a powerful architecture. However, to process collections such as ZINC [28] or Enamine [50] with hundreds of million chemical compounds requires a much more sophisticated approach.

[5] In this publication we are referring to the binary-classification task. This topology requires a slight modification for multi-class classification problems.

[6] A latent vector is a point representing the original input in the latent space. The latent space is a collection of vectors, generated by a complex compression function (*encoder*). It is extremely difficult (and more likely impossible) to demonstrate this on a practical example. This figure is an attempt to demonstrate the *variational autoencoder* concept in relation to chemical compounds. By introducing this figure we are trying to help our reader to understand how *variational autoencoder* works within the scope of this paper. If the reader wants to take a deeper dive into *variational autoencoder* theory, we recommend the following publication [33].

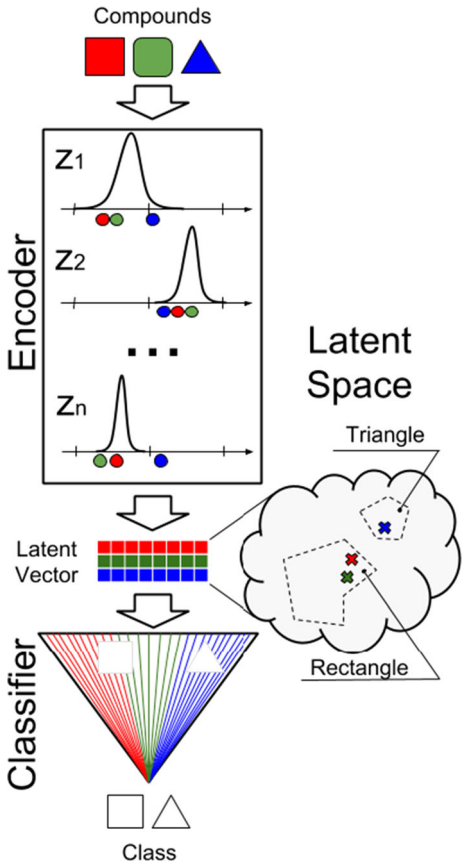


**Fig. 10** Classification example using the latent space

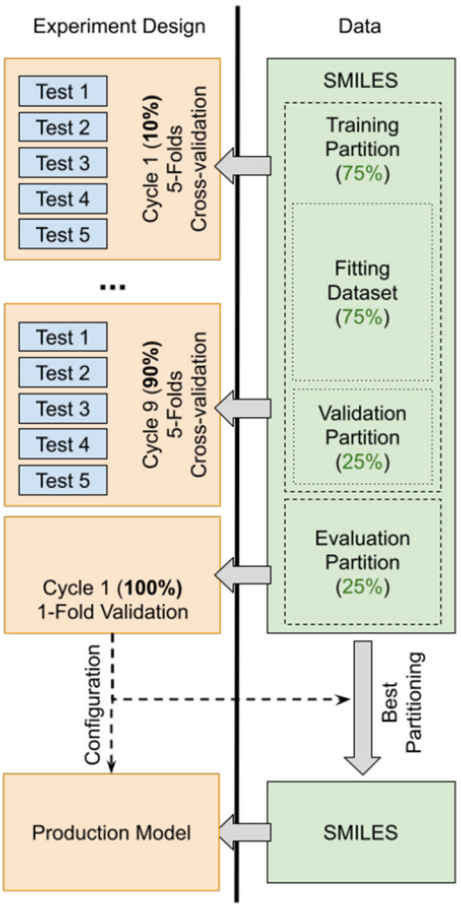


**Fig. 11** Design of experiments for *autoencoder* model

This experiment tries to define, what is an optimal size of a training data set for an accurate reconstruction of SMILES. This will help to scale down training data set and to preserve reconstruction accuracy at the same time. A design of proposed experiments is shown in Fig. 11.

All data taking part in the experiment were normalized and filtered using MolVS Open Source software [4]. SMILES exceeding 150 characters were removed. This had no significant impact on overall quality of experimental results, since only a very small percentage of these compounds were discarded. After filtering the data set comprises 1688073 samples.[7] This data set had been randomly split into training and evaluation partitions in proportions of 75% and 25% of samples respectively. The evaluation data (422,019 samples) remained unchanged throughout all experiments. It helped to score all produced models against the same benchmark. The size of training data varied from 10% to 100% of the original size (1,266,054 samples) depending on an experiment configuration. Generators were developed to feed data to a training model during a fitting processes. These generators streamed data directly from files using fixed size batches (equals to 1024), preventing any memory overflow.

Ten groups of experiments were carried out. Five tests were performed in each group, except for the last one.[8] Each test randomly selected a certain percentage of samples from the internal training partition. Then, these samples were divided into fitting and validation subsets in proportions of 75% to 25% respectively. The validation subset was used in each training epoch to assess model performance. This assessment was necessary to control learning rate, checkpoints and earlier stopping mechanisms. An approximate number of chemical compounds selected for each test is shown in Table 1.

The best setup (experiment 5) shown in Table 2 was selected to build a production *autoencoder* model (which was used in the further tests for building *classification* and *regression* models). A explanation for the selected data split (defined by the experiment 5) is provided in Sect. 5.1 and discussed in Sect. 6.

[7] A number chemical compounds used in the referenced study [21] is different, due to an outdated filtering algorithm.

[8] The last group handled 100% of data, no random sampling was needed.

**Table 1** ChEMBL data set split

| Experiment group | Percent of samples (%) | # of training samples | # of validation samples |
|---|---|---|---|
| 1 | 10 | 94,953 | 31,651 |
| 2 | 20 | 189,907 | 63,302 |
| 3 | 30 | 284,861 | 94,953 |
| 4 | 40 | 379,815 | 126,605 |
| 5 | 50 | 474,768 | 158,256 |
| 6 | 60 | 569,722 | 189,907 |
| 7 | 70 | 664,676 | 221,558 |
| 8 | 80 | 759,630 | 253,210 |
| 9 | 90 | 854,583 | 284,861 |
| 10 | 100 | 949,537 | 316,512 |

**Table 2** ChEMBL data set production split

| # of samples | Description |
|---|---|
| 474,768 | Fitting compounds |
| 158,256 | Validation compounds |
| 422,019 | Evaluation compounds |

Autoencoder model efficacy was evaluated by reconstruction accuracy, Hamming [44] and Levenshtein [42] editing distances.

## 4.2 Experimental data and setup for regression problem

LogD values for training and evaluation of a regression model were obtained from the ChEMBL database. To prevent influence outliers on the overall model only values in the interval between $LogD \in [-20, +20]$ were considered. Chemical compounds with normalization issues were also excluded during the pre-processing step. The total number of chemical compounds taking part in testing was 1669058.

A data set containing these values was obtained from ChEMBL using the following steps:

1. Data fields *smi* and *val* were retrieved from ChEMBL using the SQL statement, where 'smi' is a chemical compound SMILES and 'val' a LogD value.
2. MolVS Open Source software [4] was used to normalize each SMILES.
3. Records with $length(smi) > 150$ were removed;
4. Records with LogD values outside the specified interval $val \in [-20, +20]$ were removed;
5. All LogD values were normalized between 0 and 1.

10-fold cross-validation was conducted to assess the model performance. The design is presented in Fig. 12.

The data selected for cross-validation was split into 10 folds where nine folds (90% of data) were taken for training and one fold (10% of data) for evaluation accordingly. The data set allocated to training was also randomly split into fitting/validation partitions in the following proportion 75%/25%. The validation partition was used at each training epoch for assessing model quality. R2-score [10] metrics were recorded for each fold, and later generalized into the final result.

An additional 10-cross validation test was introduced to evaluate the regression model based on the experimental LogD data. Data were downloaded from latest ChEMBL version (on 30 Sept 2018) and pre-processed by industry experts in drug-discovery.

- The data was curated to remove results obtained with solvents other than octanol/aqueous buffer.[9]
- Results derived from HPLC retention times were also removed.[10]
- Results obtained from experiments conducted at pH other than pH7.4 were also removed (both high and low pH.[11]
- Duplicates were removed by using of InChiKeys.[12]

[9] The vast majority of the historical data available was determined using octanol/aqueous buffer as the liquid phases, there is a body of data using cyclohexane/aqueous buffer but this was excluded to reduce the number of variables involved in the experiments.

[10] Traditionally partition coefficients are determined experimentally using the Shake Flask Method [45]. Since this test can be time-consuming, alternative methods have been explored; HPLC retention times [34], however, applying a regression equation derived from one chemical class to a second one may not be reliable. For internal consistency only data from experiments using the shake flask method were used.

[11] The majority of the experiments were carried out at physiological pH (7.4) there is small amount of data from different pH but since this will affect the extent of ionization of the molecules alternative pH data was removed.

[12] Some molecules had been tested many times, to give equal weight to every molecule a single record for each molecule was required. InChiKeys were generated for each molecule [3] and used to compare records, duplicate structures were removed.

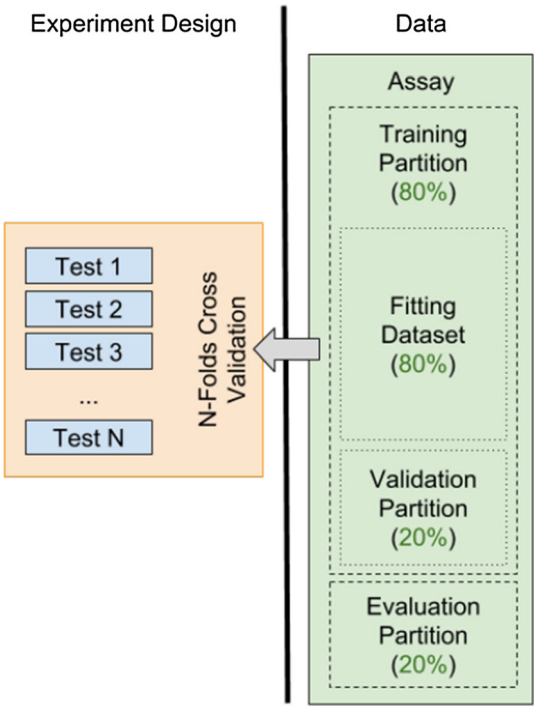


**Fig. 12** Design of experiments for predicting LogD using a regression model and drug-target binding using a classification model

After cleaning and pre-processing this data set consisted of 12413 samples which LogD property defined in the interval $-12.0, +12.0$.

The second data set 'Lipophilicity' was obtained from the ML resource described in [55]. It consists of 4200 chemical compounds which LogD property is defined in the following interval $[-1.5, +4.5]$. This data set was used to gauge the performance of developed system against other ML models.

### 4.3 Experimental data and setup for classification problem

ChEMBL[13] contains approximately 13.5M bio-activity measurements, where 1.1M assays are assigned to approximately 11K targets. The majority of available bio-activity data are highly unbalanced. More than 50% of assays have just a single measurement while others contain tens of thousands. On the other hand, a lot of targets belong only to a single assay, while others to hundreds. A large proportion of these data contain duplicate records. Such heterogeneity of data prevents clear identification, which measurements can be considered as *active* or *inactive* accordingly. A special protocol proposed in [40] helps to generate benchmark data sets for binary classification. It has the following six steps:

1. Data fields *smi*, *typ*, *unt*, *rel* were retrieved from ChEMBL using the SQL statement, where '*smi* ' field represents a SMILES, '*typ* ' a type of measurement, '*val* ' a measurement value, '*com* ' a measurement comment, '*unt* ' a measurement unit and '*rel* ' a measurement relation. These fields abbreviations are used throughout this study for referencing. All identified assays are belonged to the 'B'-type. These data are measures of compound binding to a molecular target, e.g. Ki, IC50, Kd.;
2. MolVS Open Source software [4] was used to normalize each of the SMILES.
3. Records with $length(smi) > 150$ were removed;
4. A measurement was considered as *active* if $com \in \mathbb{A}$. A measurement was considered as *inactive* if $com \in \mathbb{I}$. Sets ($\mathbb{A}$ and $\mathbb{I}$) of strings in comment-field are defined below. If a record is identified as *inactive* all further steps are discarded. $\mathbb{A}$ = ('active', 'note: corresponding ic50 reported as active') $\mathbb{I}$ =('inconclusive', 'not active', 'inactive', 'not active (inhibition $<$ 50% @ 10 um and thus dose-response curve)')
5. Removed all records passed the previous step where $val = \emptyset \mid unt \neq' \text{nM}' \mid rel \notin \{' >',' \geq',' <',' \leq',' =',' \sim'\}$.
6. Assigned labels to each record according to the defined thresholds presented in Sect. 12:

$$\text{label} = \begin{cases} 1 : val \geq 5.5 \\ 0 : val < 5.5 \end{cases} \tag{12}$$

7. All records with duplicates and contradictory measurements obtained during the previous steps were discarded.

56 assays were identified for this study. Each assay reflects *in vitro* measurements obtained during HTS. A break down between *active* or *inactive* chemical compounds together with associated target are presented in Tables 3 and 4.

Three cross-validation experiments were carried out for each data set. A design of these experiment is shown in Fig. 12. This folds number was chosen due to the relatively small assay sizes. As it can be seen from Table 5, a large proportion of assays have approximately 800 compounds. Three folds provide a fair representation of *active* or *inactive* chemical compounds across training, validation and evaluations partitions.

The Maximum Unbiased Validation (MUV) [48] is another benchmark data set selected from PubChem BioAssay by applying a refined nearest neighbour analysis. The MUV data set contains 17 challenging tasks for around 90 thousand compounds and is specifically designed for validation of virtual screening techniques. The detail breakdown between *active* and *inactive* compounds is shown in Table 5.

[13] version 23

**Table 3** Classification ChEMBL data set statistics (Part I)

| ChEMBL ID | Active count | Inactive count | Target description |
|---|---|---|---|
| 1794375 | 4473 | 78672 | Unchecked |
| 1614421 | 11,391 | 37,563 | Microtubule-associated protein TAU |
| 1614249 | 813 | 41,195 | Ferritin light chain |
| 1614166 | 98 | 34,096 | Muscleblind-like protein 1 |
| 1614364 | 1519 | 10,298 | Tyrosyl-DNA phosphodiesterase 1 |
| 3214913 | 872 | 3474 | Unchecked |
| 3215169 | 542 | 2651 | Unchecked |
| 1909170 | 52 | 736 | Muscarinic acetylcholine receptor M1 |
| 1909171 | 47 | 743 | Muscarinic acetylcholine receptor M2 |
| 1909172 | 49 | 739 | Muscarinic acetylcholine receptor M3 |
| 1909173 | 50 | 735 | Muscarinic acetylcholine receptor M4 |
| 1909174 | 55 | 739 | Muscarinic acetylcholine receptor M5 |
| 1909191 | 32 | 757 | Progesterone receptor |
| 1909209 | 55 | 735 | Serotonin 1a (5-HT1a) receptor |
| 1909211 | 79 | 705 | Serotonin 2a (5-HT2a) receptor |
| 1909085 | 59 | 735 | Alpha-1a adrenergic receptor |
| 1909086 | 62 | 730 | Alpha-1b adrenergic receptor |
| 1909087 | 63 | 732 | Alpha-1d adrenergic receptor |
| 1909088 | 81 | 700 | Alpha-2a adrenergic receptor |
| 1909089 | 80 | 705 | Alpha-2b adrenergic receptor |
| 1909090 | 59 | 723 | Alpha-2c adrenergic receptor |
| 1909094 | 74 | 725 | Norepinephrine transporter |
| 1909102 | 50 | 739 | Unchecked |
| 1909104 | 96 | 695 | Serotonin 2b (5-HT2b) receptor |
| 1909105 | 85 | 707 | Serotonin 2c (5-HT2c) receptor |
| 1909108 | 53 | 741 | Serotonin 6 (5-HT6) receptor |
| 1909109 | 68 | 718 | Serotonin transporter |
| 1909110 | 59 | 736 | Sigma opioid receptor |
| 1909111 | 42 | 748 | Unchecked |

For comparing prediction models the Receiver Operating Characteristics - Area Under The Curve (ROC-AUC) [12] was used. This metric is widely excepted by a ML community to assess performance of classification models.

## 5 Results

According to the experimental setup defined in Sect. 4, results are presented in three sub-sections. The first Sect. 5.1 shows results for *variational autoencoder*, the second Sect. 5.2 for prediction of LogD (regression problem) and the final Sect. 5.3 for prediction of compounds-targets binding (binary classification problem). The main rationale behind these experiments is to validate the versatility of latent vector based fingerprint, in other words, to prove that it works well regardless of the selected problem (classifier or regression).

### 5.1 SMILES reconstruction

Forty-six tests (9x5+1 for more details see 4.1) were conducted to assess the accuracy of SMILES reconstruction on different portion of training data. The results obtained are detailed in Table 6.

Changes in accuracy and editing distance for different size of training data sets are presented in Figs. 13 and 14. As it can be seen from Fig. 13, the reconstruction accuracy increases from $0.247 \pm 0.027$ for 10% of randomly selected samples to $0.877 \pm 0.009$ for 50% of samples accordingly. From 60% onwards accuracy stays around 0.8 on average with slight fluctuations. This is an expected result. However slight variations in accuracy starting from 60% of samples needs to be addressed. It is a difficult task to identify the exact reason of what is influencing these

**Table 4** Classification ChEMBL data set statistics (Part II)

| ChEMBL ID | Active count | Inactive count | Target description |
|---|---|---|---|
| 1909112 | 71 | 724 | Unchecked |
| 1909121 | 59 | 740 | Unchecked |
| 1909130 | 37 | 754 | Cyclooxygenase-1 |
| 1909132 | 31 | 760 | Cytochrome P450 1A2 |
| 1909134 | 40 | 752 | Cytochrome P450 2C19 |
| 1909135 | 34 | 751 | Cytochrome P450 2C9 |
| 1909136 | 56 | 726 | Cytochrome P450 2D6 |
| 1909139 | 41 | 751 | Dopamine D1 receptor |
| 1909140 | 45 | 749 | Dopamine D2 receptor |
| 1909141 | 74 | 714 | Dopamine D3 receptor |
| 1909143 | 55 | 743 | Dopamine transporter |
| 1909150 | 36 | 758 | Glucocorticoid receptor |
| 1909156 | 43 | 751 | Histamine H1 receptor |
| 1909159 | 50 | 743 | Unchecked |
| 1613896 | 640 | 1048 | Cytoplasmic zinc-finger protein |
| 1614122 | 719 | 954 | Zinc finger protein mex-5 |
| 1614192 | 446 | 1138 | Luciferin 4-monooxygenase |
| 2328568 | 857 | 160 | Sodium channel protein type IX |
| 3215187 | 415 | 441 | Unchecked |
| 3706045 | 529 | 37 | Mitogen-activated protein kinase 14 |
| 3706373 | 715 | 83 | 5-lipoxygenase activating protein |
| 3705899 | 681 | 84 | Complement factor D |
| 1614063 | 33 | 711 | Glyceraldehyde-3-phosphate dehydro. |
| 3705488 | 630 | 106 | Serine/threonine-protein kinase B-raf |
| 3705869 | 511 | 139 | Pyruvate dehydro. kinase isoform 2 |
| 3705476 | 527 | 135 | Acetyl-CoA carboxylase 2 |
| 3734213 | 101 | 482 | 6-phosphofructo-2-kinase/fructose-2 |
| 3214944 | 389 | 189 | Histone-lysine N-methyltrans. NSD2 |

changes in accuracy. One of the possible reasons for such behaviour is early signs of overfitting. Obviously there is no evidence of this phenomenon spotted during training. However, it is possible that with increasing of samples number a model starts to memorize the input. This can be addressed by a better sampling algorithm. For example, Butina clustering [13] can be a useful technique to design a sampling algorithm.

In addition to accuracy, model performance was measured using Hamming and Levenshtein distances. Both belong to a family of editing distances and give a different perspective on the results obtained. They show a similar trend to accuracy. The Hamming distance decreased from $4.374 \pm 0.817$ to $0.663 \pm 0.071$ for 10% and 50% of sample cases respectively. The Levenshtein distance decreased from $4.299 \pm 0.127$ to $0.648 \pm 0.031$ for the same percentage of samples. Both show slight fluctuation in editing distance from 60% onwards.

According to the carried out experiment the best result (accuracy $0.877 \pm 0.009$) was obtained for 50% of randomly selected samples. This configuration was selected for building a production *variational autoencoder*, which can be later used for training *classification* and *regression* models with SMILES input. Training was conducted using 40 epochs, and produced a model with accuracy *0.872*. The editing distances for the production model were *0.682* and *0.677* for Hamming and Levenshtein respectively.

## 5.2 LogD prediction

Two cross validation experiments described in Sect. 4.2 were carried out on ChEMBL and Lipophilicity data sets. Results of these experiments are presented in Table 7.

As it can be seen from the obtained results the best performance is achieved for the ChEMBL data set, with an average coefficient of determination of $0.907 \pm 0.008$. A scatter plot with an alignment of true and predicted values is shown in Fig. 15. The training process was carried out

**Table 5** Classification MUV data set statistics

| ChEMBL ID | Active count | Inactive count |
|---|---|---|
| 466 | 14,814 | 78,274 |
| 548 | 14,705 | 78,383 |
| 600 | 14,698 | 78,390 |
| 644 | 14,593 | 78,495 |
| 652 | 14,873 | 78,215 |
| 689 | 14,572 | 78,516 |
| 692 | 14,614 | 78,474 |
| 712 | 14,383 | 78,705 |
| 713 | 14,807 | 78,281 |
| 733 | 14,654 | 78,434 |
| 737 | 14,662 | 78,426 |
| 810 | 14,615 | 78,473 |
| 832 | 14,637 | 78,451 |
| 846 | 14,681 | 78,407 |
| 852 | 14,622 | 78,466 |
| 858 | 14,745 | 78,343 |
| 859 | 14,722 | 78,366 |

with LogD values normalized in the interval $[-20, +20]$[14] and all training cycles took 20 epochs.

The high prediction accuracy is a very much expected result, since the majority of LogD data points in ChEMBL are computed using the ACD/Labs software [1]. In this scenario the regression model is simply learning to predict an outcome of another computational algorithm (such as ACD/Labs). It makes a learning task much more straightforward. This assumption is vindicated by results obtained in other studies. For example experiments with ChEMBL data set described in [1] also show a high accuracy using a SVM [19] model.

Much more interesting results are obtained for the Lipophilicity data set. The average R2 score equals to $0.542 \pm 0.021$, which is noticeably less in comparison to ChEMBL data. The training process was carried out with the LogD normalized interval $[-1.5, +4.5]$ and all training cycles took 30 epochs. A longer training cycle reflects the complexity of building a predictive model on real experimental data. A scatter plot with an alignment of true and predicted values for the Lipophilicity data set is shown in Fig. 16.

Eight ML models were investigated on this data. A comparison chart for all these models is presented in Fig. 17. The highest score *0.697* is obtained for MPNN, which uses a generalized model [23]. It is very suitable for processing graph structured data, which makes it efficient in predicting properties of chemical compounds (since chemical compounds can be easily represented as an undirected graph). MPNN is closely followed by GC model [31], with R2 score equals to *0.662*. GC utilities principles of circular fingerprints described in [35] representing molecular structures by atom neighbourhoods. Similar to GC, Weave [31] (*0.636*) is another graph-based model which processes chemical compounds as a undirected graph using a convolution approach. In contrast to the previous three models, XGB [14] provides a different approach for making predictions. It is an ensemble approach which combines predictions of individual decision trees. XGB coefficient of determination for Lipophilicity equals to *0.577* and is closely followed by the model developed with in this study ($0.542 \pm 0.021$). Directed Acyclic Graph (DAG), Kernel Ridge Regression (KRR) and Random Forests (RF) are the lowest performing in this evaluation with R2 scores *0.507*, *0.496* and *0.483* respectively.

The key challenge of developing ML models for predicting the properties of chemical compounds is to encode molecules into fixed-length strings or vectors representation [54]. Despite SMILES providing unique representations of molecules, the majority of ML models are also relaid on additional information such as electronic or topological, profiles of chemical compounds. To derive these features, the models, we are gauging against, applied different factorizations: Extended Connectivity Fingerprints (ECFP), Coulomb matrix, Grid features, etc. These approaches are computationally expensive, and there will always be a trade-off between speed, accuracy and expense. Because of this, it is not surprising that some models provide better performance compared to our approach. As has been already mentioned, our approach is purely data-driven and inspired by - et al.[25]. It should provide alternatives to replace crafted featurization methods with the learning ability of DNN. In the future development, we are planning to improve the encoding method and directly compare it to existing featurization approaches.

Open-source research in AI always provides a solid benchmark for assessing in-house models. However, it would be interesting to compare the developed model against a commercial application. The following results show comparison of predictions made by ChemAxon software [5] against the developed regression model. This

[14] The deviation of predicting LogD values base on the interval $[-20, +20]$ may be seen to be high. However, predicting LogD is a particularly challenging problem since we have to predict lipophilicity (LogP) and the ionization (pKa) of a molecule. The distribution can range considerably but the hope is that by identifying outliers we can identify those structural classes that provide unique challenges to the algorithm. In some cases this will be because the structural class is not well exemplified within the training set, in other cases, it may be that very minor structural changes have very profound effects and the more coarse-grained models don't give sufficient accuracy.

**Table 6** SMILES reconstruction on different portion of training data

| % of training data | Accuracy | Hamming distance | Levenshtein distance |
|---|---|---|---|
| 10 | 0.247 ± 0.027 | 4.374 ± 0.817 | 4.299 ± 0.127 |
| 20 | 0.449 ± 0.012 | 2.560 ± 0.651 | 2.523 ± 0.083 |
| 30 | 0.747 ± 0.023 | 1.110 ± 0.464 | 1.102 ± 0.093 |
| 40 | 0.857 ± 0.007 | 0.829 ± 0.263 | 0.816 ± 0.015 |
| 50 | 0.877 ± 0.009 | 0.663 ± 0.071 | 0.648 ± 0.031 |
| 60 | 0.756 ± 0.035 | 0.962 ± 0.167 | 0.898 ± 0.236 |
| 70 | 0.816 ± 0.028 | 0.707 ± 0.082 | 0.766 ± 0.139 |
| 80 | 0.771 ± 0.014 | 0.913 ± 0.115 | 0.901 ± 0.137 |
| 90 | 0.726 ± 0.036 | 1.076 ± 0.275 | 0.998 ± 0.181 |
| 100 | 0.805 ± 0.000 | 0.911 ± 0.000 | 0.918 ± 0.000 |

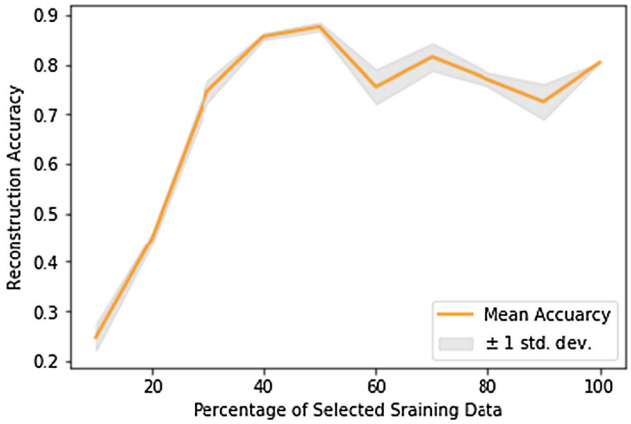


**Fig. 13** SMILES reconstruction accuracy for different percentage of training sample

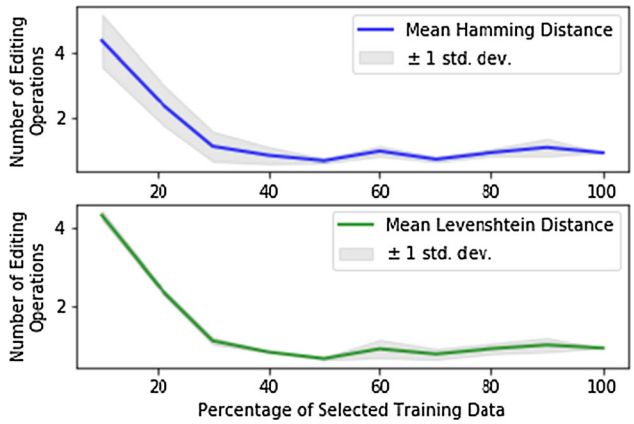


**Fig. 14** SMILES reconstruction editing distances for different percentages of training samples

work was carried out in collaboration with Cambridge MedChem Consulting [2]. Data were selected and pre-processed by industry experts in drug discovery. All undertaking steps for preparing this experiment are described in Sect. 4.2.

10-folds cross-validations were performed on selected data set to compare models.[15] The obtained results are shown in Table 8. Results for the developed regression are shown in the 'ARM' column and for the commercial software in the 'ChemAxon' column accordingly.

An average R2 score for our developed regression model equals to $0.695 \pm 0.013$, which is nearly twice that obtained by ChemAxon with R2 of $0.338 \pm 0.034$. It is hard to comment on the underlining ChemAxon algorithm for making prediction without available source code. However, from the description presented on the company website [5] it possible to assume, that it is a deterministic algorithm which approximates a chemical compound structure into a specific property value. Two scatter plots represented in Figs. 18 and 19 show an alignment of true and predicted values based on our developed regression model and ChemAxon respectively[16]

The experiments clearly demonstrate the validity of our proposed model to predict the LogD property of chemical compounds. However a large proportion of tasks required simple classification, for example whether a compound binds to the specified target. An evaluation of the classification model constructed based on *variational autoencoder* is presented in the next Sect. 5.3.

## 5.3 Binding prediction

Two cross validation experiments described in Sect. 4.3 were carried out on 56 ChEMBL data sets. Results of these experiments are presented in Table 9.

Considering the large volume of obtained results, they were split into five groups based on ROC-AUC metric (see Fig. 20). The first group combines 10.7% of assays with

[15] ChemAxon software was used only to generate predictions for the evaluation fold. Since it is not required any training, 9-folds allocated for this purposes were discarded at each iteration.

[16] Scatter plots based on one of the folds obtained during cross-validation.

**Table 7** Cross-validation results prediction of the LogD property for ChEMBL and Lipophilicity data sets

| Fold | ChEMBL | Lipophilicity |
|---|---|---|
| 1 | *0.911* | *0.563* |
| 2 | *0.904* | *0.518* |
| 3 | *0.908* | *0.574* |
| 4 | *0.917* | *0.538* |
| 5 | *0.897* | *0.554* |
| 6 | *0.904* | *0.569* |
| 7 | *0.916* | *0.526* |
| 8 | *0.906* | *0.542* |
| 9 | *0.913* | *0.512* |
| 10 | *0.889* | *0.525* |
| Summary | 0.907 ± 0.008 | 0.542 ± 0.021 |

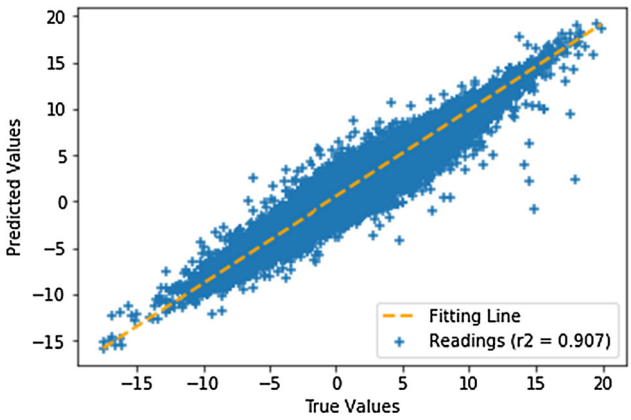


**Fig. 15** Scatter plot of predicted LogD values on ChEMBL data

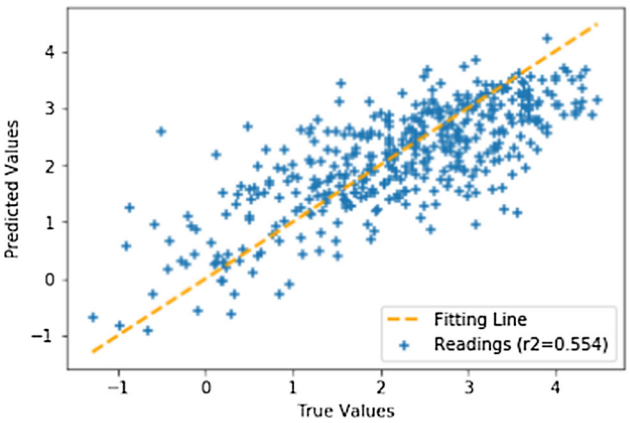


**Fig. 16** Scatter plot of predicted LogD values on Lipophilicity data

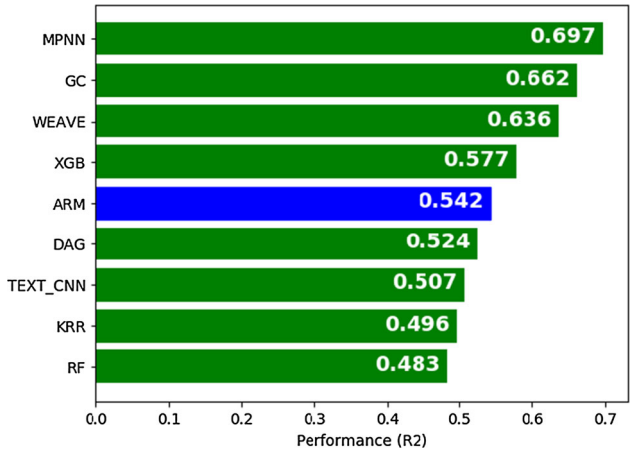


**Fig. 17** SA comparison chart for ML models of predicting Lipophilicity

least accurate prediction (which ROC-AUC is located in (0.0, 0.6) interval). It is very closely followed by the next group of 12.5% assays, which showed result in [0.6, 0.7) interval. A slightly bigger group of 17.9% of assays demonstrated ROC-AUC in interval [0.7, 0.8). In many cases such accuracy of in-silico prediction on HTS data can be already consider as a very good result. However the largest group, which combines 50% of assays showed ROC-AUC scores in interval [0.8, 0.9). Such accuracy can have a significant impact on planning and execution of HTS experiments, majority of assays filtered by in-silico approach. The remaining group combines 8.9% assays with highest scores located in interval [0.9, 1.0).

Despite competent results obtained on vast majority of tested assays, the authors carried out additional investigation to rank the developed classifier against other ML algorithms. Similar to the regression problem, the main objective here is not a direct comparison of different ML algorithms, since it requires different experimental setup. This study projects prediction accuracy observed for the developed model on results already described by - et al. [55]. The ROC-AUC characteristics for 17 binding-assays are presented in Table 10.

An average ROC-AUC of a 10-folds cross-validation experiment was recorded for each assay. The summary field represents an average ROC-AUC across all 17 experiments. It was used for ranking the developed classifier against 6 ML algorithms. A visual representation of this ranking is shown in Fig. 21.

Similar to the regression problem, GC - a graph based model, scored *0.775* the best result for binding classification.[17] It closely followed by BYPASS, representing a multitask neural network and LOGREG representing logistic regression model, scoring *0.764* and *0.749*

[17] It ranked as a second for regression problem, but a gap with the leader (MPNN) is very close.

**Table 8** Cross-validation for predicting LogD using developed regression model and ChemAxon software

| Fold | ARM | ChemAxon |
|---|---|---|
| 1 | *0.695* | *0.290* |
| 2 | *0.721* | *0.332* |
| 3 | *0.694* | *0.346* |
| 4 | *0.690* | *0.376* |
| 5 | *0.710* | *0.290* |
| 6 | *0.675* | *0.331* |
| 7 | *0.660* | *0.272* |
| 8 | *0.693* | *0.347* |
| 9 | *0.701* | *0.260* |
| 10 | *0.698* | *0.394* |
| Summary | 0.693 ± 0.016 | 0.323 ± 0.042 |

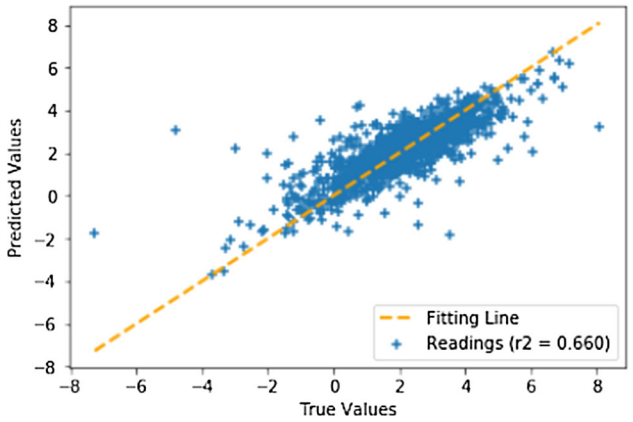


**Fig. 18** Scatter plot of predicted LogD values on in-vivo data using our model

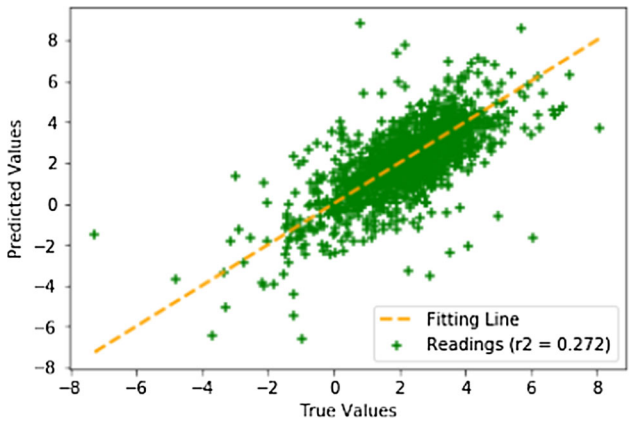


**Fig. 19** Scatter plot of predicted LogD values on in-vivo data using ChemAxon model

**Table 9** ChEMBL binding-assays with testing results

| ChEMBL ID | ROC-AUC | ChEMBL ID | ROC-AUC |
|---|---|---|---|
| 1794375 | 0.634 ± 0.003 | 1909111 | 0.832 ± 0.025 |
| 1614421 | 0.700 ± 0.007 | 1909112 | 0.846 ± 0.063 |
| 1614249 | 0.539 ± 0.043 | 1909121 | 0.844 ± 0.047 |
| 1614166 | 0.500 ± 0.000 | 1909130 | 0.851 ± 0.043 |
| 1614364 | 0.680 ± 0.011 | 1909132 | 0.741 ± 0.107 |
| 3214913 | 0.671 ± 0.013 | 1909134 | 0.674 ± 0.073 |
| 3215169 | 0.672 ± 0.025 | 1909135 | 0.766 ± 0.034 |
| 1909170 | 0.884 ± 0.035 | 1909136 | 0.729 ± 0.039 |
| 1909171 | 0.813 ± 0.014 | 1909139 | 0.816 ± 0.115 |
| 1909172 | 0.904 ± 0.030 | 1909140 | 0.785 ± 0.074 |
| 1909173 | 0.853 ± 0.039 | 1909141 | 0.838 ± 0.018 |
| 1909174 | 0.919 ± 0.008 | 1909143 | 0.754 ± 0.030 |
| 1909191 | 0.976 ± 0.006 | 1909150 | 0.982 ± 0.002 |
| 1909209 | 0.813 ± 0.036 | 1909156 | 0.917 ± 0.039 |
| 1909211 | 0.871 ± 0.043 | 1909159 | 0.713 ± 0.046 |
| 1909085 | 0.854 ± 0.056 | 1613896 | 0.517 ± 0.003 |
| 1909086 | 0.842 ± 0.040 | 1614122 | 0.597 ± 0.021 |
| 1909087 | 0.854 ± 0.008 | 1614192 | 0.848 ± 0.013 |
| 1909088 | 0.821 ± 0.051 | 2328568 | 0.751 ± 0.035 |
| 1909089 | 0.805 ± 0.071 | 3215187 | 0.515 ± 0.014 |
| 1909090 | 0.856 ± 0.010 | 3706045 | 0.800 ± 0.086 |
| 1909094 | 0.815 ± 0.026 | 3706373 | 0.712 ± 0.023 |
| 1909102 | 0.819 ± 0.012 | 3705899 | 0.839 ± 0.005 |
| 1909104 | 0.861 ± 0.017 | 1614063 | 0.776 ± 0.012 |
| 1909105 | 0.851 ± 0.019 | 3705488 | 0.656 ± 0.053 |
| 1909108 | 0.823 ± 0.055 | 3705869 | 0.801 ± 0.029 |
| 1909109 | 0.834 ± 0.024 | 3705476 | 0.544 ± 0.067 |
| 1909110 | 0.806 ± 0.061 | 3734213 | 0.635 ± 0.011 |

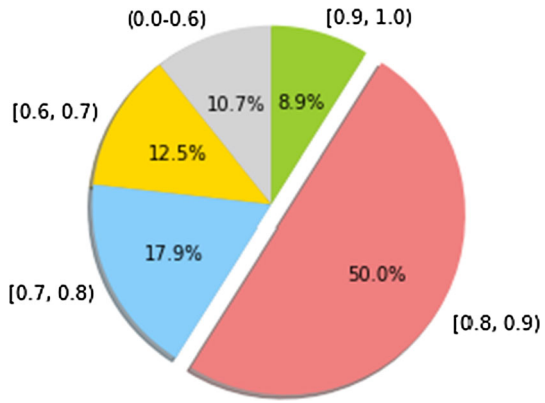


**Fig. 20** Binding prediction results split into five groups based on the ROC-AUC metric

respectively. XBG model showed *0.720* ROC-AUC score which was closely followed by ARM, with *0.696*. Influence Relevance Voting (IRV) systems and Random Forests (RF) are the poorest performing, with ROC-AUC scores of

**Table 10** MUV binding-assays with testing results

| MUV ID | ROC-AUC | MUV ID | ROC-AUC |
|---|---|---|---|
| 466 | 0.841 ± 0.005 | 733 | 0.648 ± 0.011 |
| 548 | 0.821 ± 0.005 | 737 | 0.602 ± 0.073 |
| 600 | 0.603 ± 0.060 | 810 | 0.656 ± 0.006 |
| 644 | 0.685 ± 0.002 | 832 | 0.604 ± 0.038 |
| 652 | 0.753 ± 0.113 | 846 | 0.815 ± 0.002 |
| 689 | 0.622 ± 0.074 | 852 | 0.662 ± 0.065 |
| 692 | 0.631 ± 0.012 | 858 | 0.651 ± 0.007 |
| 712 | 0.570 ± 0.098 | 859 | 0.797 ± 0.011 |
| 713 | 0.870 ± 0.006 | Summary | 0.696 ± 0.035 |

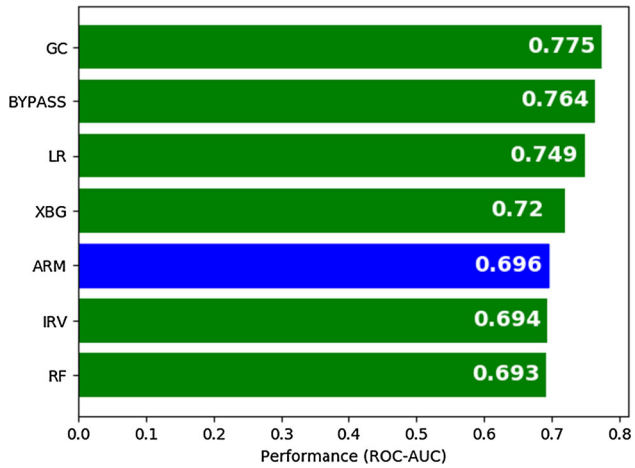


**Fig. 21** Ranking the developed classifier against six ML models using binding prediction

*0.693* and *0.693* respectively. Despite variations in accuracy all techniques showed a very consistent performance. Considering the hugely unbalanced data sets the measured metric can be significantly shifted by producing one extra true positive prediction.

With such variety of different models capable of producing equally accurate results for regression and classification problems, why another approach? This question is addressed in the below.

## 6 Discussion and conclusion

During the last few years deep neural networks de facto have become an industry standard for creating sophisticated AI models. A significant amount of effort have been devoted to improve classical ML algorithms. Often XBG and RF produce even better results than DNNs. Such a variety of approaches makes it a very difficult task to choose the right technique. This has also made a huge impact on scientific publications. Researchers are forced to show rigorous testing, with comparison of every proposed technique against the recognized leaders. It expected that the developed technique must outperform others, which in practice leads to compromising with experiments design and cherry-picking phenomenon.

It can be seen from our comparison of published models, the majority show very similar performance. However, it is also important to score each model in term of practical application. For example a model can deliver impressive accuracy, but when it needs to be deployed in a production environment, scalability and efficiency diminish all advantages gained in perfecting the quality of predictions.

This work has been inspired by a research effort of using *variational autoencoder* to generate chemical compounds fingerprints, using these fingerprints for predicting specific properties of chemical compounds. Considering the high complexity of chemical compounds to train quality *variational autoencoder* requires a large data set of SMILES. A typical size of HTS assays consists of several hundred, maybe thousands of chemical compounds. Such volume of data samples does not deliver a sufficient variety of chemical compounds structures, which makes it impractical to train a *variational autoencoder* based on assay data.

A decision was made to use ChEMBL data to obtain a large representation of chemical compounds structures. It worked very well, with the developed *variational autoencoder* model reconstructing nearly 90% of SMILES using 1024 latent vector. Taking into the account Hamming and Levenshtein editing distances the final model in average has one misplaced or incorrect atom or bond. Obviously such error is unforgivable in chemistry, but the primary objective of *variational autoencoder* is to produce latent space where similar structures are crumbled together. Let's demonstrate this on simple example.

Assume that the selected target chemical compound is `NC1=NNC(=C1)C1=CC(F)=CC(F)=C1`, which structure is shown Fig. 22.

The search space for similar chemical compounds was reduced to 10,000 samples (out of 674,040 initially allocated for evaluation, for more detail see Sect. 4.1) to make computation more efficient. K-nearest neighbour [36] identified five closest chemical compounds defined in Table 11, and illustrated in Fig. 23.

The last column in Table 11 represents Tanimoto similarity score [9]. It is widely used in the chemistry domain to assess similarity between two chemical compounds. Two compounds can be considered similar if Tanimoto score is greater than 0.85 (for Daylight fingerprints). As it can be seen from the obtained results in Table 11 three compounds retrieved using latent space are also similar

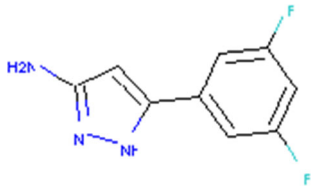

**Fig. 22** 2D Structure of chemical compound `NC1=NNC(=C1)C1=CC(F)=CC(F)=C1`

**Table 11** Definitions of five closest chemical compounds retrieved for the specified `NC1=NNC(=C1)C1=CC(F)=CC(F)=C1` target

| # | SMILES | Tanimoto |
|---|---|---|
| 1 | `NC1=NNC(=C1)C1=CC(Cl)=CC(Cl)=C1` | *0.94* |
| 2 | `NC1=NN(CC2=C3N=CC=CC3=CC=C2)C=C1` | *0.54* |
| 3 | `NC1=NNC(=C1)C1=C(Cl)C=CC(Br)=C1` | *0.87* |
| 4 | `NC1=NN(C=C1)C1=C(Cl)C=CC=N1` | *0.71* |
| 5 | `NC1=NNC(=C1)C1=CC(Br)=CC(F)=C1` | *0.97* |

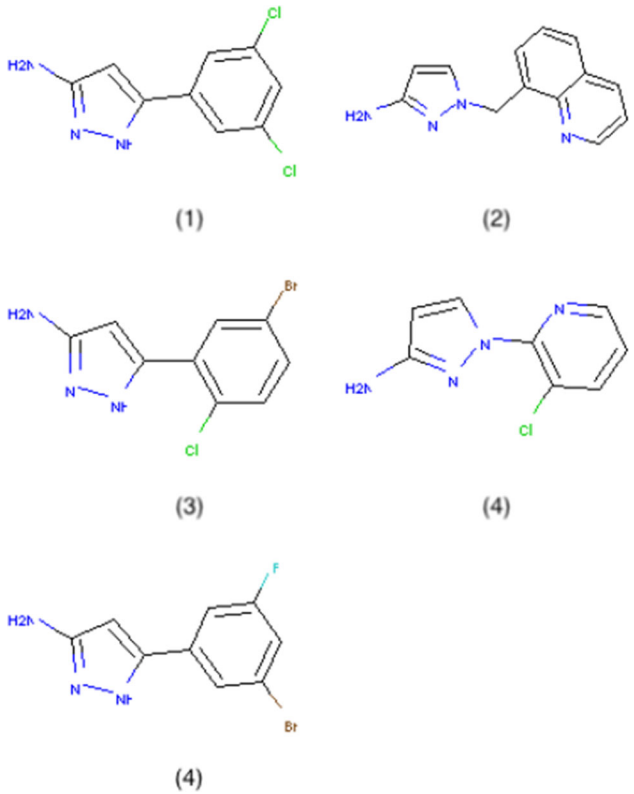


**Fig. 23** Structures of five closest chemical compounds retrieved for the specified `NC1=NNC(=C1)C1=CC(F)=CC(F)=C1` target (the structure number corresponds to the compound number in Table 11

according to Tanimoto metric, where the other two results are closely followed.

This example demonstrates that latent vector based fingerprints can be used to define similarity between two chemical compounds. It also clearly shows that selected chemical compounds are closely located in latent space.

The closest approximation of similar compounds in the latent space provides potential capability for the developed model to be applied to generating new chemical compounds and forecasting the desired properties. Such ML models become a hot topic in pharmaceutical domain, which can be witnessed by increasing the number of high-quality research publications in this space [25, 46]. In this paper, the main focus was on the transfer learning, to use the trained encoder as a base for classification or regression networks which can predict properties of chemical compounds. However, the decoder can be potentially used for generating novel chemical compounds. By introducing a small modification into the latent representation of a target chemical compound, it is possible to generate a novel structure. Despite this simple idea the implementation of such a model is very complex and outside the scope of this publication.

The trained *variational autoencoder* forms a solid base for creating different *classification* and *regression* models. An interesting pattern was observed during a training process. The majority of trained models converge to a stale state (where no longer improvement observed) during 2-3 epochs. However when the same topology of neural network was trained without pretrained *variational autoencoder*, the training process continue up to 100 epochs. Longer training is not a problem for relatively small data set, where full learning cycle can be complied in the matter of hours. However in case of such collection volume as ChEMBL, training may go on for days. Also, the experiments did not reveal any degradation in accuracy with shortening the training cycle.

The question is why are *classification* and *regression* models built based on *variational autoencoder* so efficient in the training process? To answer this question, let us come back to the methodology described in Sect. 3. Constructed *classification* and *regression* models consists from two parts. The first part is an *encoder* isolated from a trained *variational autoencoder*. The second part is MLP, which performs actual predictions. Effectively, the MLP is trained to make predictions based on chemical compound fingerprints. Since the 'hard work' has been already done by *variational autoencoder*, only a few cycles are required to learn differentiation rules (to solve classification or regression problem).

A further observation which was called 'latent space drift' was also noted. A number of publications using a similar approach do not clearly reveal their mechanisms of using an *encoder*. It can be used with frozen and unfrozen layers. Using an *encoder* with frozen layers makes a lot of

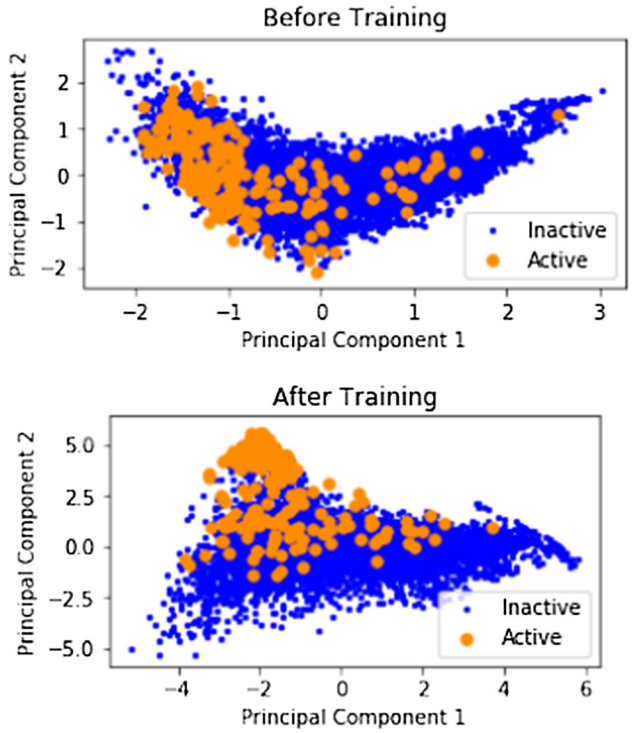


**Fig. 24** An example of the latent space drift. The diagrams show a distributions of training points of binary classes before and after the drift

sense. If it is already trained to encode SMILES, then the attached layers can only be trained to utilize the obtained chemical compounds fingerprints. However, a preliminary study showed that if an *encoder* is left unfrozen, the training result is generally better. An initial investigation found that fingerprint points in latent space become adjusted according to the target (predicting) property. This process is explained in Fig. 24.

All this internal analysis of neural network behaviour becomes possible due the specialized AUROMIND software. It provides a set of tools for 'debugging' a training process. The authors plan to describe some of the core principles behind developed tools in upcoming publications.

**Acknowledgements** We acknowledge the contribution of Chris Swain the Founded Cambridge MedChem Consulting.

## Declarations

**Conflicts of interest** The authors declare that they have no conflict of interest.



## References

1. Acd/labs software. https://www.acdlabs.com. Accessed 16 Jul 2019
2. Cambridge medchem consulting provides a range of consultancy services in drug discovery and medicinal chemistry. https://www.cambridgemedchemconsulting.com. Accessed 16 Jul 2019
3. International chemical identifier. https://en.wikipedia.org/wiki/International_Chemical_Identifier. Accessed 30 Jan 2021
4. MolVS molecule validation and standardization. https://molvs.readthedocs.io/en/latest/. Accessed 16 Jul 2019
5. Software solutions and services for chemistry & biology. https://chemaxon.com. Accessed 16 Jul 2019
6. Aghdam HH, Heravi EJ (2017) Guide to convolutional neural networks: a practical application to traffic-sign detection and classification, 1st edn. Springer Publishing Company Incorporated, Berlin
7. Agrawal A, Choudhary A (2016) Perspective: Materials informatics and big data: realization of the fourth paradigm of science in materials science. APL Materials 4:053208
8. Agrawal A, Deshpande P, Cecen A, Gautham B, Choudhary A, Kalidindi S (2014) Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. Integr Mater Manuf Innov 3:90–128
9. Bajusz D, Rácz A, Héberger K (2015) Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? J Cheminform 7(1):20
10. Bartels R (2015) Re-interpreting r-squared, regression through the origin, and weighted least squares
11. Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, Krüger FA, Light Y, Mak L, McGlinchey S et al (2014) The chembl bioactivity database: an update. Nucleic Acids Res 42(D1):D1083–D1090
12. Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159
13. Butina D (1999) Unsupervised data base clustering based on daylight's fingerprint and tanimoto similarity: a fast and automated way to cluster small and large data sets. J Chem Inf Comput Sci 39(4):747–750
14. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD '16, ACM, New York, NY, USA, pp 785–794
15. Cox B, Merritt AT, Binnie A, Donnelly MC, Mander TH, Denyer JC, Evans B, Green DV, Lewis JA, Valler MJ, Watson SP (2000) 3-application of high-throughput screening techniques to drug discovery. Elsevier, Amsterdam, pp 83–133
16. Davies M, Nowotka M, Papadatos G, Dedman N, Gaulton A, Atkinson F, Bellis L, Overington JP (2015) Chembl web services: streamlining access to drug discovery data and utilities. Nucleic Acids Res 43(W1):W612–W620
17. Dietterich T (1995) Overfitting and undercomputing in machine learning. ACM Comput Surv 27(3):326–327

18. Duvenaud DK, Maclaurin D, Aguilera-Iparraguirre J, Gómez-Bombarelli R, Hirzel T, Aspuru-Guzik A, Adams R (2015) Convolutional networks on graphs for learning molecular fingerprints. CoRR arXiv:1509.09292
19. Evgeniou T, Pontil M (2001) Support vector machines: theory and applications. pp 249–257. https://doi.org/10.1007/3-540-44673-7_12
20. Gagorik AG, Savoie B, Jackson N, Agrawal A, Choudhary A, Ratner MA, Kohlstedt KL (2016) Improved scaling of molecular network calculations: the emergence of molecular domains. J Phys Chem Lett 8:415–421
21. Galushka M, Browne F, Mulvenna MD, Bond R, Lightbody G (2018) Toxicity prediction using pre-trained autoencoder. In: IEEE international conference on bioinformatics and biomedicine, BIBM 2018, Madrid, Spain, December 3–6, pp 299–304
22. Garciarena U, Santana R, Mendiburu A (2018) Expanding variational autoencoders for learning and exploiting latent representations in search distributions. In: Proceedings of the genetic and evolutionary computation conference, GECCO '18, ACM, New York, NY, pp 849–856
23. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. CoRR arXiv:1704.01212 (2017)
24. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning, vol 1. MIT Press Cambridge
25. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, Aguilera-Iparraguirre J, Hirzel TD, Adams RP, Aspuru-Guzik A (2018) Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Sci 4(2):268–276
26. Heaton J, Polson N, Witte JH (2016) Deep learning in finance. arXiv preprint arXiv:1602.06561
27. Hoffman MD, Blei DM, Wang C, Paisley J (2013) Stochastic variational inference. J Mach Learn Res 14(1):1303–1347
28. Irwin J, Shoichet B (2005) Zinc - a free database of commercially available compounds for virtual screening. J Chem Inf Model 45:177–82
29. Simplified S (2014) Normal distribution. J Conserv Dent 17(1):96–97
30. Jiang X, Zhang, Y., Zhang, W., Xiao, X.: A novel sparse auto-encoder for deep unsupervised learning. In: 2013 Sixth international conference on advanced computational intelligence (ICACI) (2013)
31. Kearnes SM, McCloskey K, Berndl M, Pande VS, Riley P (2016) Molecular graph convolutions: moving beyond fingerprints. J Comput Aided Mol Des 30(8):595–608
32. Ker J, Wang L, Rao J, Lim T (2018) Deep learning applications in medical image analysis. IEEE Access 6:9375–9389
33. Kingma DP, Welling M (2019) An introduction to variational autoencoders. CoRR arXiv:1906.02691 (2019)
34. Klose M, Theiner S, Varbanov H, Hoefer D, Pichler V, Galanski M, Meier-Menches S, Keppler B (2018) Development and validation of liquid chromatography-based methods to assess the lipophilicity of cytotoxic platinum(iv) complexes. Inorganics 6(4):130. https://doi.org/10.3390/inorganics6040130
35. Koutsoukas A, St Amand J, Mishra M, Huan J (2016) Predictive toxicology: modeling chemical induced toxicological response combining circular fingerprints with random forest and support vector machine. Front Environ Sci 4:11
36. Kramer O (2013) K-nearest neighbors. Springer, Berlin, p 2013
37. Lusci A, Pollastri G, Baldi P (2013) Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. J Chem Inf Model 53(7):1563–75
38. MacKay DJC (1998) Introduction to monte carlo methods. In: Jordan MI (ed) Learning in graphical models, NATO science Series, Kluwer Academic Press, Amsterdam, pp 175–204 (1998)
39. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D (2014) The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp 55–60
40. Mayr A, Klambauer G, Unterthiner T, Steijaert M, Wegner JK, Ceulemans H, Clevert DA, Hochreiter S (2018) Large-scale comparison of machine learning methods for drug target prediction on chembl. Chem Sci 9:5441–5451
41. Menard S (2002) Applied logistic regression analysis. No. v. 106; v. 2002 in Quantitative applications in the social sciences. Sage Publications, New York
42. Miller FP, Vandome AF, McBrewster J (2009) Levenshtein distance: information theory, computer science, string (computer science), string metric, damerau? Levenshtein distance, spell checker, hamming distance. Alpha Press, Orlando
43. Mozaffar M, Paul A, Al-Bahrani R, Wolff S, Choudhary A, Agrawal A, Ehmann K, Cao J (2018) Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks. Manuf Lett 18:35–39. https://doi.org/10.1016/j.mfglet.2018.10.002
44. Norouzi M, Fleet DJ, Salakhutdinov RR (2012) Hamming distance metric learning. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems, vol 25, pp 1061–1069. Curran Associates, Inc
45. OECD: Test No. 107: Partition Coefficient (n-octanol/water): Shake Flask Method (1995). https://doi.org/10.1787/9789264069626-en
46. Popova M, Isayev O, Tropsha A (2018) Deep reinforcement learning for de novo drug design. Sci Adv 4(7):eee7855
47. Raiber F, Kurland O (2017) Kullback-leibler divergence revisited. In: Proceedings of the ACM SIGIR international conference on theory of information retrieval, ICTIR '17, ACM, New York, NY, pp 117–124
48. Rohrer SG, Baumann K (2009) Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. J Chem Inf Model 49(2):169–184
49. Ruder S (2017) An overview of multi-task learning in deep neural networks. CoRR arXiv:1706.05098(2017)
50. Shivanyuk A, Ryabukhin S, Bogolyubsky A, Mykytenko D, Chuprina A, Heilman W, Kostyuk A, Tolmachev A (2007) Enamine real database: making chemical diversity real. Chim Oggi 25:58–59
51. Swamidass SJ, Azencott CA, Lin TW, Gramajo H, Tsai SC, Baldi P (2009) Influence relevance voting: an accurate and interpretable virtual high throughput screening method. J Chem Inf Model 49(4):756–766
52. Weininger D (1988) Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28(1):31–36
53. Wishart D, Knox C, Guo A, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J (2006) Drugbank: a comprehensive resource for in silico drug discovery and exploration. Database Issue 34:668–672
54. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V (2018) Moleculenet: a benchmark for molecular machine learning. Chem Sci 9:513–530

55. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande VS (2018) Moleculenet: a benchmark for molecular machine learning. Chem Sci 9(2):513–530
56. Zhang C, Ma Y (2012) Ensemble machine learning: methods and applications. Springer, New York
57. Zhang Y, Duchi J, Wainwright M (2015) Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. J Mach Learn Res 16(1):3299–3340