Chapter 12

Integrated machine learning framework for computer-aided chemical product design

Qilei Liu, Haitao Mao, Lei Zhang, Linlin Liu, and Jian Du
Institute of Chemical Process System Engineering, School of Chemical Engineering, Dalian University of Technology, Dalian, China

1 Introduction

Modern society requires many chemical-based products for its survival, such as fuels, plastics, detergents, medicines, and food [1]. As the increasing requirements for sustainable and human-centered developments, chemical product design has been paid wide attentions from the scientific communities and chemical industries. Traditionally, chemical products are designed and developed through heuristic rule-based and/or trial-and-error experiment-based approaches. Although these kinds of approaches often lead to safe and reliable product designs, it is not practically feasible to evaluate all alternatives or to obtain the optimal solution. Recently, the use of model-based design methods has been gaining increased attention as they have the potential to generate and/or screen feasible product candidates in a much larger design space, and at the same time, reduce the time and costs for their development [2]. If required data and the models giving reliable estimations for product properties and functions are available, it is possible to develop and use model-based chemical product design approaches with the advances in computer-aided technologies. During last three decades, efforts have been made to develop databases, design methods and associated software tools for chemical products. Review articles for the design methods of chemical products can be found in Zhang et al. [1–3]. Although good progress has been made in model-based approaches for chemical product design, due to the multidisciplinary and multiscale nature of chemical products, challenges still exist which impede the development and application of chemical product design methodologies. The multidisciplinary and multiscale nature of chemical product design makes it difficult to understand basic scientific issues of chemical products, which hinders the

Applications of Artificial Intelligence in Process Systems Engineering
https://doi.org/10.1016/B978-0-12-821092-5.00004-8

establishment of reliable and quantitative physicochemical property models. Thus, these hurdles prevent the use of model-based approaches for the design of chemical products. Nowadays, with the development of the data-driven methods, machine learning (ML) has been regarded as an alternative solution to the above-mentioned challenges.

ML has experienced successful resurgence during the last decades. The explosive growth of data with sophisticated ML algorithms make it possible to establish the ML models, which show promising potentials in applications of chemistry and chemical industry [5, 6]. Nowadays, the fundamental paradigm of statistical analysis has changed from system identification to predictive modeling. ML can model complex chemical properties due to its abilities in autonomously learning data characteristics and trends [7]. Because of the adoptable generalization power, the established ML can be used in more general scenarios and thus has abilities in designing chemical products. Moreover, increasing efforts have been made to interpret ML models, for obtaining the insights of ML from basic scientific issues to chemical products. As a result, ML is possible to be one of the major research trends in chemical product design. Nowadays, ML methods have been widely applicated in different aspects of chemical product design. One of the most important roles of ML methods is establishing the quantitative structure-property relationships (QSPRs). Growing number of new potentially useful ML methods in chemistry, such as the use of artificial neural networks (ANNs) [8], in quantitative structure-activity relationships (QSARs) [9] and in ligand-based virtual screening [10]. The ML ability to predict key properties of a product has been highlighted by Zonouz et al. [11] who developed an ANN coupled with a genetic algorithm for the modeling and optimization of toluene oxidation over perovskite-type nano-catalysts. In the area of crystallization, Velásco-Mejía et al. [12] employed ANN combined with a genetic algorithm in modeling and optimization of process design. Liu et al. [13] outlined the typical modes and basic procedures for applying ML in materials science. Pankajakshan et al. [14] provided conceptual scheme to obtain chemical insights into complex phenomena and development of predictive models for material design. Similarly, Vanhaelen et al. [15] emphasized ML-based workflow in drug design.

In this chapter, our recent works on ML-based chemical product design (focus on molecular product design) are presented. In Section 2, an overall ML-CAMD framework is presented. In Section 3, the ML-CAMD framework is discussed in detail for the establishment of ML models for property prediction as well as chemical product design. In Section 4, two case studies are presented for the applications of the proposed framework.

2 An integrated ML framework for computer-aided molecular design

The structure-property relationships are essential in property prediction and chemical product design as these relationships serve as the “bridge” between

the molecular structure/product constitution and the desired property. In this section, an overall ML-CAMD framework is presented to assist in the establishment of the ML models for the missing structure-property relationships for property prediction and chemical product design. The diagrammatic sketch of the ML-CAMD framework is shown in Fig. 1, which consists of four steps: data collection, data preprocessing and feature engineering, model establishment, and chemical product design.

In the data collection step, a set of chemical products (molecules, mixtures, blends, etc.) as well as their product descriptors (groups, compositions, etc.) and known properties from experiments, literatures, etc. are organized as a database for the establishment of the ML models. Then, the established database is sent to the data preprocessing and feature engineering step to eliminate the abnormal values, complement the missing data and avoid high computation load and overfitting issues in model establishment. Finally, the ML model is established for property prediction through model selection, training and validation, as well as chemical product design through efficient mathematical optimization algorithms.

3 Establishment of ML model for computer-aided molecular design

In this section, the steps in the proposed ML-CAMD framework are applied to establish the ML models for computer-aided molecular design (CAMD). These steps are discussed in detail in the following content.

3.1 Data collection

Create database. It is essential to establish a database before establishing ML models. Many molecular databases have already been available online, such as NIST Chemistry webbook (www.webbook.nist.gov/chemistry/), PubChem (www.pubchem.ncbi.nlm.nih.gov), ZINC (www.zinc.docking.org), DRUGBANK (www.drugbank.ca), ChEMBL (https://www.ebi.ac.uk/chembl/), and ProCAPD database (www.pseforspeed.com/procapd), etc. For the design of different types of products, separate databases with their separate ontologies are required. For examples, separate databases are needed for solvents, aroma compounds, active ingredients for different types of functional products, refrigerants, membranes, adsorbents, and many more. Therefore, the elements (molecules, ingredients, etc.) need to be carefully selected in the database to establish a balanced tailor-made ML model between generalization and accuracy for a fixed type of product.

Select product descriptors. The data in the database contain descriptors (e.g., groups for molecules, ingredient mole fractions for mixtures, bonding-based graphical representation for crystals). Descriptor is one of the key factors to determine the performance of ML models. The selection of descriptors varies

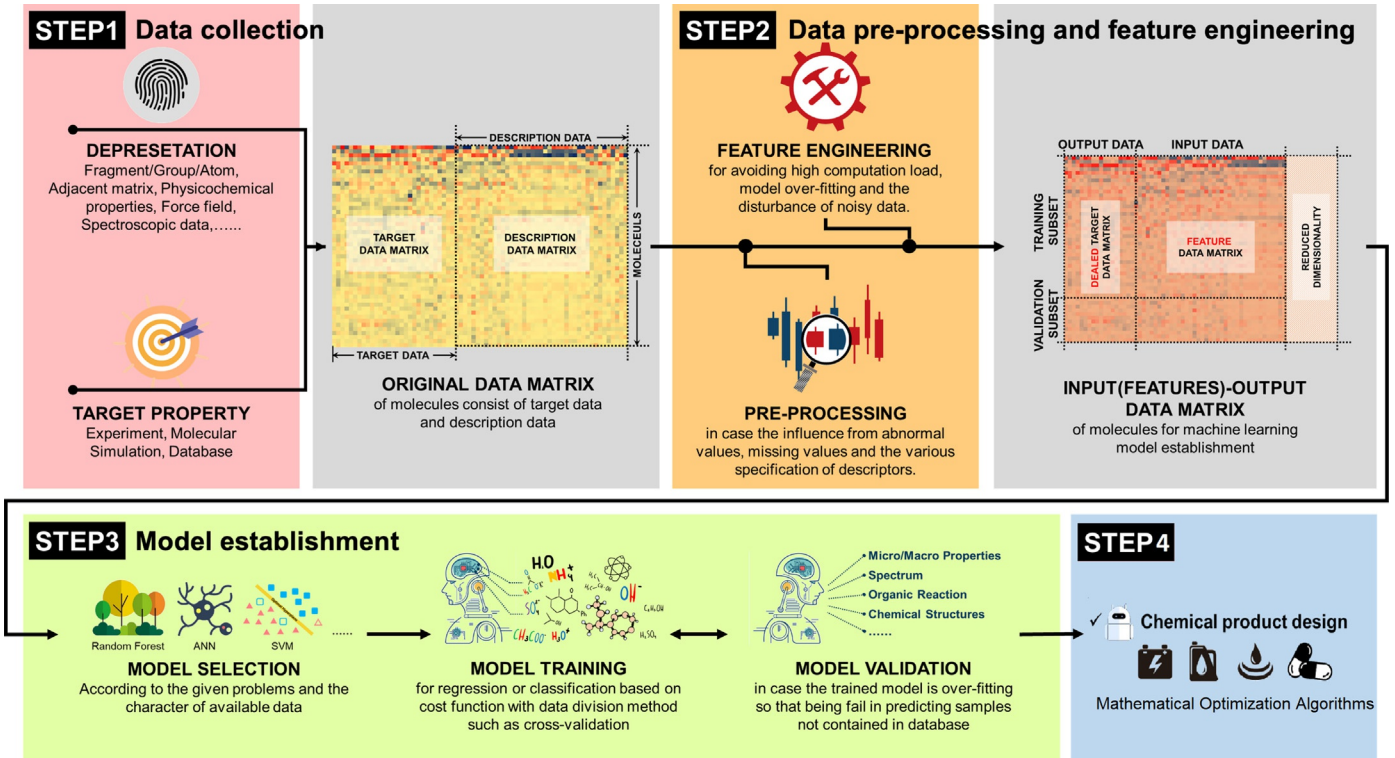


FIG. 1 The diagrammatic sketch of the ML-CAMD framework.

TABLE 1 Commonly used descriptors in chemical product design problems.

Categories	Descriptors	Application examples
Practical values	• Experimental physicochemical descriptors • …	• Design of catalysts [16]
Groups	• Functional groups (CH_3, CH_2, OH, …) • Adjacency matrix • …	• Prediction of pH [17] • Estimation of cetane and octane numbers [18] • Odor prediction [4]
Chemoinformatic	• Constitutional descriptors • Topological descriptors • Physicochemical descriptors • Structural descriptors • …	• Drug design [19] • Material design [20]
Molecular fingerprints	• Hashed fingerprint [21] • Extended-connectivity fingerprints [22] • …	• Organic chemistry reaction prediction [23]
Molecular spectrum	• IR, UR • NMR, Raman • Material image • …	• Material property prediction [24]
Computational chemistry	• Initial nuclear velocities • Chemical environment • Molecular orbitals • …	• Catalyst prediction [25] • Ab initio molecular dynamics simulations [26] • Predict chemical reactions [27]
Graph convolution	• 2D molecular graph • …	• Drug efficacy and photovoltaic prediction [28] • Properties of different biological classes [29]
Text description	• SMILES [30] • SMARTS [21] • …	• Drug design [31]

with the product types and design problems. Table 1 shows some commonly used descriptors in chemical product design problems.

The advantages and disadvantages of different descriptors are discussed as follows:

- *Practical values*: The advantage is that practical values are accurate to measure molecular physicochemical properties, and thereby leading to

reliable ML-based property prediction models. The disadvantage is that practical values are not easily accessible.

- *Groups*: The advantage is that groups are structural descriptors, which is easily obtained with a minor computational cost. However, the definitions of reasonable groups are highly dependent on expert knowledge.
- *Chemoinformatic*: The advantage is that chemoinformatic has a large number of structural and property descriptors and is easily available through theoretical calculations. The disadvantage is that chemoinformatic descriptors have the issue of inconsistent dimensions, which may result in poor training results for ML models.
- *Molecular fingerprints*: The advantage is that fingerprints contain molecular topological information and is easily available through theoretical calculations. The disadvantage is that fingerprints are 2-dimensional descriptors, and they cannot correlate 3-dimensional properties, for example, molecular docking calculations between drugs and target proteins.
- *Molecular spectrum*: The advantage is that molecular spectrum is able to represent molecular features and integrate with graph neural networks. The disadvantage is that molecular spectra are not easily accessible.
- *Computational chemistry*: The advantage is that computational chemistry descriptors are pseudo experimental data and is obtained by theoretical calculations. However, the high computational cost for computational chemistry descriptors hinders the high-throughput design and/or selection of chemical products.
- *Graph convolution*: The advantage is that graph convolution is able to summarize the local chemical environments of atoms and intramolecular connectivity, which is suitable for establishing graph neural networks. The disadvantage is that they have difficulties in representing stereoscopic characteristics of molecules.
- *Text description*: The advantage is that text descriptors have a strong expansibility and wide applicability for ML modeling. The disadvantage is that text descriptors (e.g., SMILES) cannot represent conformational isomers.

Collect product properties. The data in the database include product properties (e.g., normal boiling point, solubility parameter, odor, color). The property data can be collected from experiments, literatures, and molecular databases. If it is hard to perform experiments or the data are unavailable in the literatures/databases, computation/simulation tools are alternatives to complementing the missing data.

3.2 Data preprocessing and feature engineering

With the accumulation of historical data and the development of computation/simulation tools, the data available for ML is often sufficient. However, these data are not always valid when directly applied in the establishment of ML

models. On the one hand, systematic errors (noises) are inevitable during experiments. On the other hand, issues such as inconsistent dimensions, inconsistent order of magnitude, high dimensionality, and irrelevant or redundant data often makes poor training results of the ML models. Therefore, it is necessary to employ data preprocessing and feature engineering methods to process the original data. Note that ML also includes deep learning without feature engineering, which is not further discussed in this chapter. More details about deep learning can be found in LeCun's work [32].

Data preprocessing is essential for the experimental data (errors are often caused by human or equipment, as well as disturbances from the environment) and the issues of inconsistent dimensions and inconsistent order of magnitude. For chemical product design problems, one of the issues is the required descriptors and/or the product properties are not always available in the database. The abnormal and missing data of the descriptors and/or properties can be replaced by the mean, median, or the most frequent values based on different preprocessing methods. These data preprocessing methods are summarized in Table 2.

Feature engineering is an indispensable technique to identify the irrelevant, redundant, and noisy data from the raw data (high dimensional data) and convert them to new features (low dimensional data), which retain the prominent characteristics of a system and contribute to the efficient learning process of ML models. For chemical product design problems, if the collected molecular descriptors (raw data) are high dimensional, it is necessary to perform feature engineering before ML modeling. Feature engineering generally consists of two methods, namely feature extraction and feature selection.

Feature extraction, for example, principal components analysis (PCA), is able to extract critical information from the original feature space (raw data) and thereby reduce the data dimension. This technique is commonly used in image processing [33] and speaker recognition [34]. For chemical product design problems, feature extraction is popular with spectra recognitions, for example, proteomic mass spectra [35] and nuclear magnetic resonance [36], as spectra are suitable descriptors to represent unique molecular structures. After feature extraction, the significant features in high dimensional spectra are identified and further associated with product properties through ML modeling.

Different from feature extraction, feature selection is a process to remove irrelevant, redundant, and noisy descriptors from the original feature space and adopt the rest as essential features for ML modeling. This technique is more promising for the estimation of product properties as the selected descriptors retain physical significance, which leads to an interpretability of ML models. Feature selection generally has three categories: (1) Filter: select descriptors as features without any ML involved; (2) Wrapper: employ a ML model to evaluate the selected features; (3) Embedding: combine the feature selection and the training process of ML model. Detailed advantages and disadvantages of the above three techniques and their corresponding specific methods are summarized in Table 3.

TABLE 2 Summary of data preprocessing methods.

	Function	Description	Advantage	Disadvantage	Formula
Standardization scaler	Nondimensionalization	Scale using the mean and standard values of one descriptor.	Remove the dimension and retain the original distribution characteristics of the data.	N/A	$x^* = {}^{x-\overline{x}}/_{s}$ $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ $s^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2$
Interval scaler	Nondimensionalization	Scale based on the maximum and minimum values of one descriptor.	Scale flexibly according to the data.	Affected if the distribution is not uniform severely.	$x^* = \frac{x - \min}{\max - \min}$ $\max = \mathrm{Max}\{x_1, x_2, \ldots, x_n\}$ $\min = \mathrm{Min}\{x_1, x_2, \ldots, x_n\}$
Normalization	Compute the similarity among samples	Calculate the *p*-norm for each sample and scale the data of a sample by dividing the corresponding norm.	A uniform standard is obtained by processing data of samples within the same row into unit vector.	The difference of dimensions in units of descriptors remains.	$x^* = \frac{x}{(\sum_i x_i^p)^{\frac{1}{p}}}$
Binarization	Process the continuous descriptors into the categorical values for classification.	Transform the values of descriptors into binary variety.	N/A	N/A	$x^* = \begin{cases} 1, & x > threshold \\ 0, & x \leq threshold \end{cases}$

Encoding categorical features	Process the descriptors not being given as continuous values but categorical.	Transform categorical features with n possible values into n binary features.	N/A	The dimension might be enormous if the values of descriptors are widely distributed.	$\boldsymbol{X} = [x_1 \ \ldots \ x_n]$ $\boldsymbol{X}^* = \boldsymbol{I}_{n\times n} = f([x_1 \ \ldots \ x_n])$
Generating polynomial features	Add complexity to the model by considering nonlinear features of the input data.	Calculate the descriptors' high-order and interaction terms as a supplement.	More feasible for modeling complexity relationship linking descriptors and properties.	Higher computational complexity during modeling the machine learning.	$\boldsymbol{X} = [x_1, x_2]$ $\boldsymbol{X}^* = [1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$
Inferring them from the known part of the data	Imputation of missing values	Replace missing values, either using the mean, the median or the most frequent value of the descriptor the missing values are located.	Enable the model to be formulated without the hindrance from missing data.	Affected by the distribution of data severely.	N/A

TABLE 3 Advantages and disadvantages of filter, wrapper, and embedding techniques and their specific methods.

Technique	Filter		Wrapper		Embedding	
Common advantage	√ Easily implement with a certain evaluation measure. √ Flexibly control the result using the criteria. √ Independent of the classifier.		√ Interacts with the learning machine. √ Model feature dependencies.		√ Better computationally complexity than wrapper. √ Interacts with the learning machine. √ Model feature dependencies.	
Common disadvantage	× Ignoring the interaction with the model. × Fail to distinguish the redundant descriptors.		× Computationally intractable. × Overfitting risk. × Model dependent selection.		× Model dependent selection.	
Specific method	• Univariate	• Multivariate	• Deterministic	• Randomized	• Nested subset methods	• Direct objective optimization
Detailed advantage	✓ Fast and Scalable	✓ Models feature dependencies	✓ Less computationally intensive ✓ Less overfitting risk	✓ Less prone to local optima	✓ Less computationally complexity	✓ Less computationally intensive
Detailed disadvantage	× Fail to distinguish the redundant descriptors	× Slower and less scalable than univariate.	× Prone to local optima	× Higher overfitting risk × Higher computationally intensive	× Higher computationally intensive	× Higher computationally complexity

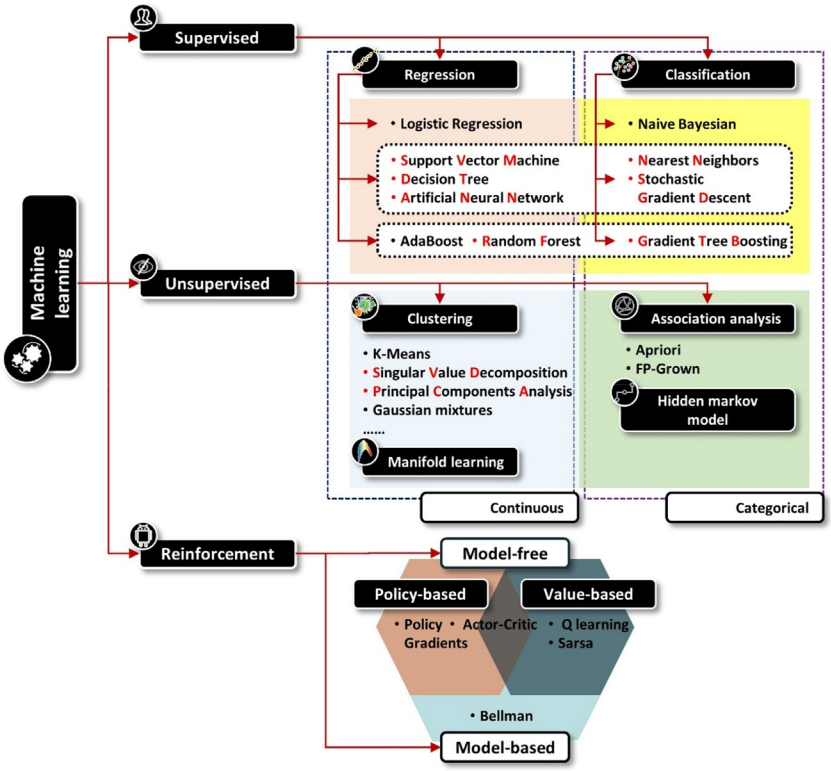


FIG. 2 The classification architecture of ML models and their application scenarios.

3.3 Model establishment

After data collection, preprocessing, and feature engineering, the features are generated and prepared for the establishment of ML model, which includes model selection, model training, and validation process.

Model selection. The model selection depends on the application scenarios, which are generally categorized into supervised learning, unsupervised learning and reinforcement learning (RL). Fig. 2 shows the commonly used ML models and their application scenarios.

Supervised learning is employed to solve two problems of regression and classification, while unsupervised learning is used to solve the problems such as clustering and association analysis. Note that the purpose of unsupervised learning is similar to dimensionality reduction, which aims to discover the potential relationships among variables. RL is the process of training the model through sequences of state-action pairs, observing the rewards that result, and adapting the model predictions to those rewards using policy iteration or value

iteration until it accurately predicts the best results. For chemical product design problems, supervised learning is a preferred method for the estimations of product properties as both continuous and discrete properties are predicted by the descriptors through regression and classification methods, respectively. Thus, supervised learning is further discussed in the following.

Currently, three major ML algorithms of supervised learning, support vector machine (SVM), decision tree (DT)/random forest (RF), and ANN, are popular with scientific research and industry practice. These algorithms have been proved possessing excellent abilities in prediction and classification. Table 4 lists several improvements and wide applications of the above ML algorithms of supervised learning.

Model training and validation process. After the type of supervised learning algorithms is confirmed, it could be built/trained by following pseudo Eqs. (1), (2).

$$\min f_{loss}(\mathbf{p}^{pre}, \mathbf{p}^{tar}) \tag{1}$$

$$\mathbf{p}^{pre} = F(\mathbf{D}, \mathbf{P}) \tag{2}$$

where f_{loss} is the loss function to quantify the difference between the prediction outputs $\mathbf{p}^{pre}$ and the target outputs $\mathbf{p}^{tar}$ (e.g., mean squared error (MSE) for prediction problems or cross entropy (CE) for classification problems), $\mathbf{D}$ is the input dataset (i.e., generated features after data preprocessing and feature engineering), $\mathbf{P}$ are the set of hyperparameters (e.g., number of hidden layers in ANN) and parameters (e.g., weights and biases in ANN) for the ML model, and F is the optimization algorithm (e.g., adaptive moment estimation (Adam) algorithm [61]) which enables the model to "learn" the relationships between inputs and outputs. The training process is actually to employ F to minimize f_{loss}. Here, the used dataset is called the training dataset.

A diagrammatic sketch of establishing a ML model is shown in Fig. 3. A well-trained ML model is always unacceptable if large prediction errors are identified among new samples, which is called generalization error or overfitting. To avoid this problem, the original dataset is divided into three subsets, namely training, validation, and testing datasets. Cross-validation methods (e.g., K-folds cross-validation) [67] are usually employed to select the training and validation datasets.

The training dataset is used to train the model and tune the ML parameters (e.g., weights and biases in ANN). If the trained ML model has poor predictions on the validation dataset, hyperparameters need to be adjusted based on knowledge or systematical methods (e.g., the grid method or the random method [7]) to prevent the overfitting problem. During the training process, introducing dropout [68] and regularization [69] layers also contributes to

TABLE 4 Abstract of typical supervised learning algorithms: remarkable improvement, advantages, disadvantages and applications.

Algorithm	Remarkable improvement	Advantage	Disadvantage	Application
Support vector machine	1. Soft margin [37]. 2. Support vector regression [38]. 3. Kernel function [39]. 4. Multiclass classification [40].	√ Separating linear indivisible problem. √ The global optimal can be found. √ Good at the classification of small sample.	× Difficult to solve problems with large. × Hard to determine a suitable kernel function. × SVM performances poor in multiclass problem.	• Prediction of viscosity [41]. • Classification of fragrance properties [42]
Decision tree/ random forest	1. Information entropy [43]. 2. Gini index [44]. 3. Pruning [45]. 4. Multivariate decision tree [46]. 5. Random tree [47].	**Decision tree** √ Comprehensible. √ Easy to construct. **Random forest** √ The error rate is significantly reduced. √ Less risk to overfitting.	**Decision tree** × Liable to be overfitting otherwise less accuracy. × Possible combination of features explosively increases. **Random forest** × Complex and time-consuming to construct. × Less intuitive.	**Decision tree** • Rate constant prediction [48]. **Random forest** • Microkinetics models [49]. • Prediction of toxicity [50]

Continued

TABLE 4 Abstract of typical supervised learning algorithms: remarkable improvement, advantages, disadvantages and applications—cont'd

Algorithm	Remarkable improvement	Advantage	Disadvantage	Application
			For both × It is limited by the application.	
Artificial neural network	**Categories of neural network** 1. Radial basis function network [51]. 2. Recurrent neural network [52]. 3. Extremely learning machine [53]. 4. Deep belief network [54]. 5. Transfer learning [55]. 6. Convolution neural network [56].	√ It is able to approximate any function, regardless of its linearity. √ Great for complex/abstract problems. √ It is able to significantly out-perform other models when the conditions are right (lots of high quality labeled data). √ Being robust to an unstable environment due to its adaptability. √ Capable of dealing big data and extracting features, based on which a high-generalization model is formulated.	× Unsuitable in cases where simpler solutions like linear regression would be best. × Requires a spate of training data. × Increasing accuracy by a few of percent need to bump up the scale by several magnitudes. × Computationally intensive and expensive. × No uniform cognition about how to configure the model or tune the parameters. × Hard to interpret the model though the data of each neuron layer could be obtained.	**Prediction to mixture** • Ternary mixture [62] • Binary mixtures [63] **Design and optimization of chemical product** • Catalysts [16] • Crystallization process [12] **Identification the structure of chemicals** • Crystal [64] • Material [14] • Catalyst [65] **Prediction to the properties involving interaction between molecules**

	7. Generative adversarial networks [57]. **Techniques for neural network** 1. Regularization 2. Back-propagation algorithm [52]. 3. Bagging [58]. 4. Dropout [59]. 5. RMSProp [60]. 6. Adam [61].			• Catalyst [66] • Partition coefficient **Predict bio-chemical properties** • Fragrance [3, 4]

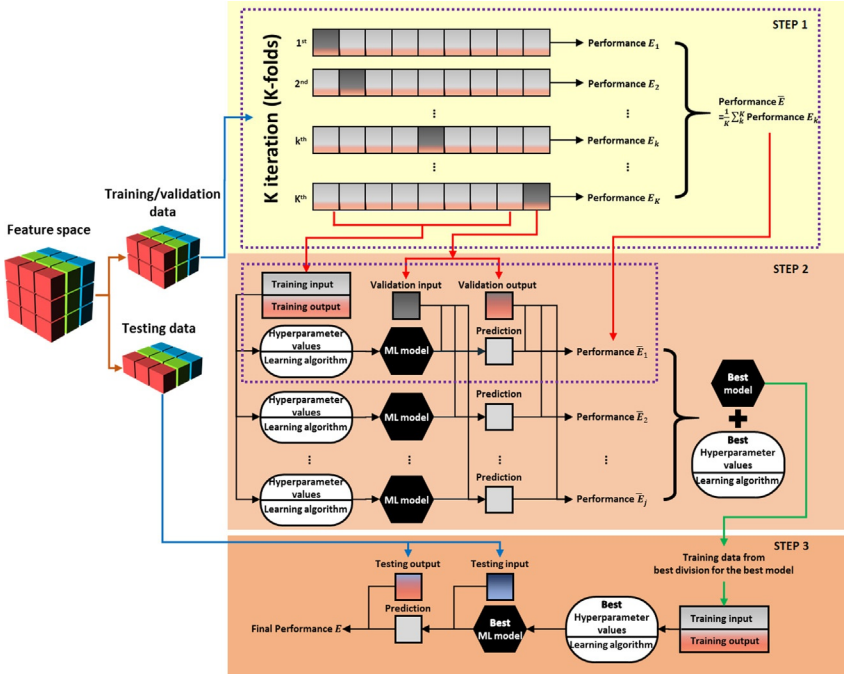


FIG. 3 A diagrammatic sketch of establishing a ML model.

avoiding the overfitting problems. Afterwards, the testing dataset is used to make a final test on the prediction accuracy and generalization ability of the ML model.

3.4 Chemical product design

In this chapter, the chemical product design problems focus on the CAMD problems. With the obtained ML models, the CAMD problem is formulated as a mixed-integer nonlinear programming (MINLP) model as follows:

$$\max/\min_{n_i(i\in\{G\})} F_{obj}(n_i)\ (\text{according to specific CAMD problems})$$

Subject to

- Molecular structure constraints: octet rule, valence bond rule and complexity constraints (linear part).

$$f(n_i)=0,\quad n_i\in\mathbb{N}^+ \tag{3}$$

- Chemical product property constraints: GC-based properties T_m, T_b, δ, $-\log(LC_{50})FM$, μ, etc. that are calculated by the GC methods (linear part).

$$P_k^L \le p_k(n_i) \le P_k^U \tag{4}$$

- Structural descriptor conversion constraints: convert the functional group sets to other molecular representations (e.g., SMILES, fingerprints, etc.) (nonlinear part).

$$f_{conv}\left(n_i, mol_{rep}\right) = 0 \tag{5}$$

- Chemical product property constraints: ML-based properties that are predicted by the ML models (nonlinear part).

$$p_{k,ML} = f_{ML}\left(mol_{rep}\right) \tag{6}$$

The ML models generally consist of nonlinear equations, which make it difficult to search for the optimal solution with other constraints simultaneously. So, a decomposition-based solution strategy [70] is used to decompose the MINLP model into an ordered set of four subproblems:

Subproblem (1): Molecular structure constraints and GC-based property constraints are first restricted to design a certain number (N_1) of feasible molecular candidates (group sets) from an ergodic combination of all functional groups by mathematical programming method.
Subproblem (2): Structural descriptor conversion constraints are then used to generate N_2 molecular representations (e.g., SMILES naming molecules) based on group sets.
Subproblem (3): Based on the molecular representations, ML models are employed to fast predict ML-based chemical product properties.
Subproblem (4): Rank the designed products based on the objective function, and a portion of top products are further verified by database, rigorous models, and/or experiments.

4 Case studies

4.1 A ML-based atom contribution methodology for the prediction of charge density profiles and crystallization solvent design

This case study refers to our previous work [83]. In this work, an optimization-based ML-CAMD framework is established for crystallization solvent design, where a novel ML-based atom contribution (MLAC) methodology is developed to correlate the weighted atom-centered symmetry functions (wACSFs) with the atomic surface charge density profiles (atomic σ-profiles, $p_{atom}(\sigma)$) using a high-dimensional neural network (HDNN) model (a kind of ML model), successfully leading to a high prediction accuracy in molecular σ-profiles ($p(\sigma)$) and an ability of isomer identification. Then, the MLAC methodology is integrated with the CAMD problem for crystallization solvent design by formulating and solving a MINLP model, where model complexities are managed with a decomposition-based solution strategy.

- **Data collection**

Create database. A $p_{atom}(\sigma)$ database is prepared for the construction of the HDNN model, where 1120 solvents containing H, C, N, O elements are collected from the Virginia Tech database [71]. Note that the samples are atoms in each solvent.

Select product descriptors. The 3-dimensional atomic descriptors, wACSFs [72], are employed to establish the HDNN model for $p_{atom}(\sigma)$ predictions. The wACSFs represent the local atomic environment of a centered atom i via the functions of radial (G_i^{rad}) and angular (G_i^{ang}) distributions of the surrounding atoms inside a cutoff sphere, which is able to describe the complex intramolecular interactions. Besides, the wACSFs are numerically calculated from the stereoscopic cartesian coordinates, and therefore are able to identify isomers (specifically, all constitutional and cis-trans isomers). More detailed information about the wACSFs can be found in Gastegger et al.'s work [72].

Collect product properties. The $p_{atom}(\sigma)$ of all solvents (product properties) are prepared with the Gaussian 09W software (http://www.gaussian.com/) and the conductor like screening model—segment activity coefficient (COSMO-SAC) model [73]. The reason for selecting $p_{atom}(\sigma)$ as the HDNN output rather than molecular $p(\sigma)$ is that the number of output samples needs to be consistent with the number of input samples (i.e., atoms) when establishing ML models. Finally, a database of $p_{atom}(\sigma)$ is established as the outputs of HDNN model, where the number of samples for H, C, N, O atoms is 15,535, 9108, 305, and 1215, respectively. In this work, the solid-liquid equilibrium (SLE) behavior in cooling crystallization process is predicted with the solvent property activity coefficients γ, which is further predicted by the COSMO-SAC model using the solvent $p(\sigma)$ (a linear addition of $p_{atom}(\sigma)$ in each solvent).

- **Data preprocessing and feature engineering**

Inconsistent order of magnitude exists among the functions of radial (G_i^{rad}) and angular (G_i^{ang}) distributions in the wACSFs, which makes poor training results of the ML models. Therefore, a standardization method is used for data preprocessing. Considering that the wACSFs are numerically calculated from the stereoscopic cartesian coordinates and are all significant in representing the local atomic environments, there is no need to employ feature engineering techniques to identify the irrelevant, redundant and noisy data from the wACSFs.

- **Model establishment**

With the obtained input (wACSFs) and output data ($p_{atom}(\sigma)$), a HDNN model (a kind of ML model) is established.

Model selection.

The HDNN model is made up of four separate element-based (H, C, N, O) ANNs. ANN is selected to correlate the wACSFs with $p_{atom}(\sigma)$ due to the following reasons:

- √ ANN has a strong ability to fit complex nonlinear relationships among data, which provides an opportunity to correlate the wACSFs with $p_{atom}(\sigma)$ since their relationships are complex, and traditional linear/nonlinear fitting methods usually fail to account for such relationships.
- √ ANN is also an efficient surrogate model with high-throughput calculation speed, which is suitable for an efficient ML-CAMD framework for solvent design.

Model training and validation process.

The molecular $p(\sigma)$ is a sum of $p_{atom}(\sigma)$ that are predicted by the HDNN model. A diagrammatic sketch of HDNN model is shown in Fig. 4. In each ANN model, the optimizer, loss function, metrics function, and activation function are Adam [61], MSE, coefficient of determination R^2 and ReLu [74], respectively. For each element (e.g., H element), atom samples are randomly divided into the training, validation, and test sample, the radios of which are 6:1:1. The size of the input vector (wACSFs) and output vector ($p_{atom}(\sigma)$) are 152 and 51, respectively. To ensure $p_{atom}(\sigma)$ nonnegative, a ReLu activation function ($f(x) = \max(0,x)$) is added to the output layer. Dropout layers are also added to the hidden layers to overcome the overfitting problem in the training process [68]. The hyperparameters (hidden layer number, neuron number in each layer, epochs, batch size, dropout ratio) are determined (as shown in Table 5) by the empirical knowledge to ensure each element-based ANN model possess good generalization (extrapolation) ability while keeping concise. The HDNN model is established on the Keras platform [75] using the Python language [76].

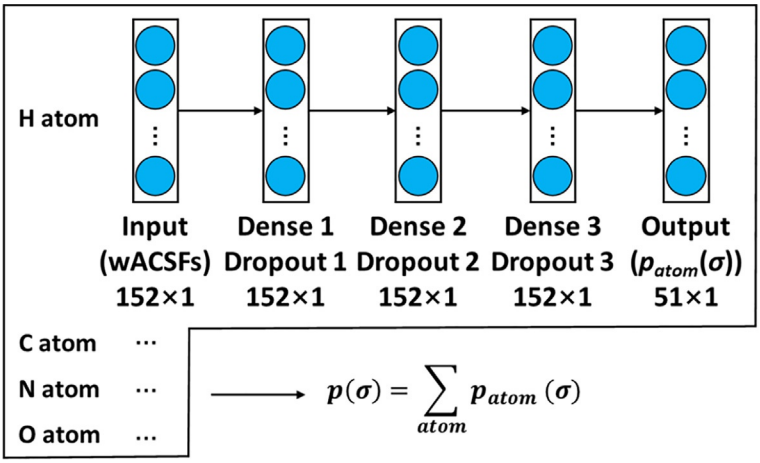


FIG. 4 A diagrammatic sketch of HDNN model.

TABLE 5 The hyperparameters of HDNN model.

Element	Sample size (training/ validation/ test)	Hidden layer number	Neuron number in each layer	Epochs	Batch size	Dropout ratio
H	11,653/ 1941/1941	3	152	1000	2000	0.05
C	6832/1138/ 1138	3	152	1000	1000	0.05
N	229/38/38	3	152	500	20	0.2
O	913/151/ 151	3	152	500	100	0.2

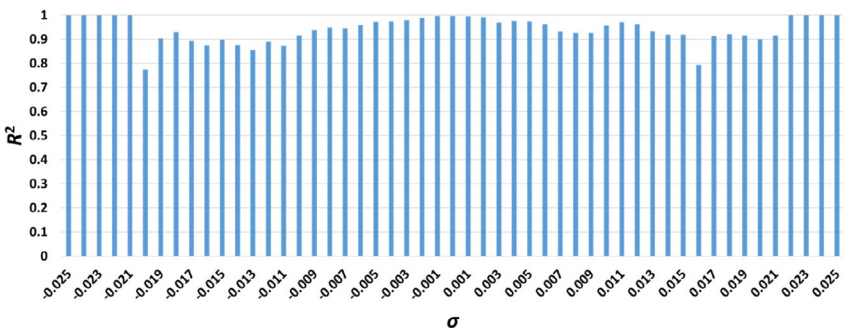


FIG. 5 The R^2 results of the predicted $p(\sigma)$ for each discrete σ interval in the MLAC method.

Finally, the metrics of training sample R^2_{train}, validation sample R^2_{val} and test sample R^2_{test} are 0.964, 0.918, 0.907 for H element, 0.975, 0.931, 0.931 for C element, 0.950, 0.889, 0.865 for N element, 0.935, 0.867, 0.902 for O element. All these results satisfy the fitting criterion $\frac{R^2_{train}-R^2_{test}}{R^2_{train}} < 0.1$ ($\frac{R^2_{train}-R^2_{test}}{R^2_{train}} \geq 0.1$ indicates overfitting) [77], indicating that the ANNs for H, C, N, O elements are reliable for $p_{atom}(\sigma)$ predictions.

To further demonstrate the feasibility and effectiveness of the HDNN model, the differences of predicted $p(\sigma)$ between the MLAC method and the density functional theory (DFT) method (benchmark) are evaluated with the criterion R^2. The R^2 results of the predicted $p(\sigma)$ for each discrete σ interval in the MLAC method are shown in Fig. 5.

Furthermore, the MLAC method is employed to provide molecular $p(\sigma)$ for the predictions of infinite dilution activity coefficients $\gamma^{\infty} = f(p(\sigma), V_C)$ based on the COSMO-SAC model, where the molecular cavity volume V_C is predicted by the group contribution (GC) method using the MG1 group sets [78] with the fitting result $R^2 = 0.9998$. The γ^{∞} predictions of the MLAC method are compared with those predicted by the DFT calculated $p(\sigma)$ and V_C. Sixteen solutes (denoted as "c1 ~ c16") and 1120 solvents that composed of H, C, N, O elements from the Virginia Tech database (solvents are classified into 13 categories and denoted as "s1 ~ s13") are selected for γ^{∞} calculations. The average absolute percent error (AAPE) criterion is used to evaluate the differences of predicted γ^{∞} between the DFT (benchmark) and MLAC method, as shown in Eq. (7).

$$AAPE = \frac{1}{T}\sum_{t=1}^{T}\frac{\left|\gamma_t^{\infty,est} - \gamma_t^{\infty,DFT}\right|}{\left|\gamma_t^{\infty,DFT}\right|} \times 100\% \tag{7}$$

where $\gamma_t^{\infty,\ est}$ is the estimated infinite dilution activity coefficients using the MLAC method (generate $p(\sigma)$) and the GC method (generate V_C), $\gamma_t^{\infty,\ DFT}$ is the DFT calculated infinite dilution activity coefficients, and T is the total number of data points. The smaller AAPE indicates the better prediction ability. The AAPEs among 16 types of solutes and 13 types of solvents using the MLAC method are shown in Fig. 6.

It is shown that most of the AAPEs in Fig. 6 are acceptable with minor prediction errors. Also, the overall AAPE for the total number of 17,920 data points (1120 solvents × 16 solutes) is calculated using the MLAC method and the result 6.6% confirms that the developed MLAC methodology is feasible and reliable to provide molecular $p(\sigma)$ (a sum of $p_{atom}(\sigma)$ that are predicted by the HDNN model) to the COSMO-SAC model for the predictions of γ^{∞}.

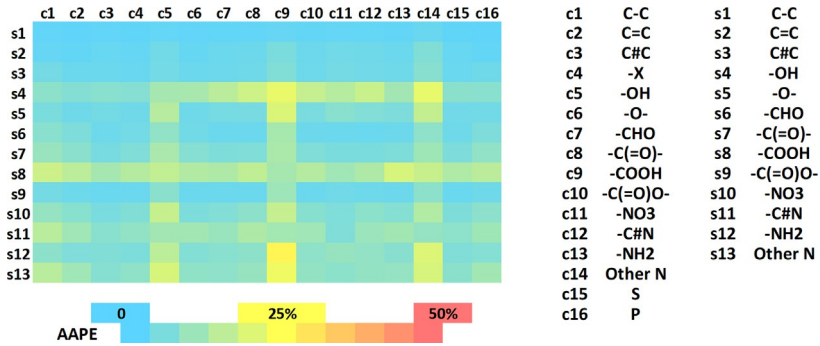


FIG. 6 The heap map of AAPEs among 16 types of solutes and 13 types of solvents using the MLAC method.

TABLE 6 List of structure and property constraints for crystallization solvent design in ibuprofen cooling crystallization process.

Property	Constraint
Number of groups	$2 \leq N_G \leq 8$
Number of same groups	$N_S \leq 8$
Number of functional groups	$1 \leq N_F \leq 8$
Hildebrand solubility parameter at 298 K	$17 \leq \delta \leq 19$ MPa$^{1/2}$
Hydrogen bonding solubility parameter	$\delta_H \geq 8$ MPa$^{1/2}$
Flash point	$T_f \geq 323$ K
Toxicity	$-\log(LC_{50})FM \leq 3.3 - \log(\text{mol/L})$
Normal melting point	$T_m \leq 270$ K
Normal boiling point	$T_b \geq 340$ K
Viscosity	$\mu \leq 1$ cP
Solid-liquid equilibrium (SLE)	$\ln x_i^{Sat} - \frac{\Delta H_{fus,i}}{RT_{m,i}}(1 - T_{m,i}/T) + \ln\gamma_i^{Sat} = 0$ $\Delta H_{fus,\ Ibuprofen} = 27.94$ kJ/mol $T_{m,\ Ibuprofen} = 347.6$ K
Molar fraction normalization	$x_1 + x_2 = 1$
Crystallization temperature range	$260 \leq T \leq 320$ K
Objective function (crystallization case study)	$PR\% = \frac{100}{1-X_L}\left(1 - \frac{X_L}{X_H}\right)$

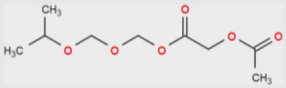
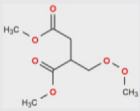
More details about the above constraints can be found in Karunanithi's work [79].

- **Chemical product design**

With the obtained HDNN model (ML model), the CAMD problem of designing a cooling crystallization solvent for Ibuprofen is formulated as a MINLP model. The following groups are selected: CH_3, CH_2, CH, C, OH, CH_3CO, CH_2CO, CHO, CH_3COO, CH_2COO, HCOO, CH_3O, CH_2O, CH—O, COOH, COO. The lower and upper bonds for structure and property constraints are given in Table 6. The objective for this case study is to design a crystallization solvent with the highest potential recovery $PR\%$.

Through using the decomposition-based algorithm, $N_1 = 272$ feasible molecular candidates (group sets) are obtained at the first step. Then, $N_2 = 6723$ SMILES-based isomers are generated using the SMILES-based isomer generation algorithm (a kind of structural descriptor conversion algorithm). After that, among N_2 solvent candidates and Ibuprofen solute, $PR\%$ are individually calculated and arranged in descending order with the key property γ_i^{Sat} that

TABLE 7 The top two designed crystallization solvents for the Ibuprofen cooling crystallization process.

SMILES	1. CC(=O)OCC(=O)OCOCOC(C)C	2. COOCC(CC(=O)OC)C(=O)OC
Molecular structure		
PR% (MLAC method)	95.91%	95.91%
PR% (DFT method)	96.04%	95.84%
δ (MPa$^{1/2}$)	18.470	18.902
δ_H (MPa$^{1/2}$)	11.018	11.329
T_f (K)	400.172	387.476
$-\log(LC_{50})FM$ ($-$log(mol/L))	2.939	2.770
T_m (K)	266.773	268.948
T_b (K)	523.919	514.742
μ (cP)	0.577	0.394

is estimated by the MLAC method (generate $p(\sigma)$, a sum of $p_{atom}(\sigma)$ that are calculated by the HDNN model), GC method (generate V_C) and COSMO-SAC model. Finally, the top two designed crystallization solvents are given in Table 7.

Although the DFT-based *PR*% of the best designed **solvent 1** in Table 7 (96.04%) has made a minor improvement (1.15% = (96.04%–94.95%)/94.95% × 100%) compared with Karunanithi's solvent (94.95%) [79], our best designed **solvent 1** is safer (T_f = 400.172 K) and lower toxic ($-\log(LC_{50})FM = 2.939 - \log(\text{mol/L})$) than Karunanithi's one (T_f = 354.290 K and $-\log(LC_{50})FM = 3.040 - \log(\text{mol/L})$). Further experimental verifications will be performed in the future to confirm the rationality of the top two designed solvents.

4.2 A ML-based computer-aided molecular design/screening methodology for fragrance molecules

This case study refers to our previous work [4]. In this work, an optimization-based ML-CAMD framework is developed for the design of fragrance

molecules, where the odor of the molecules are predicted using a data-driven ML approach, while a GC-based method is employed for prediction of important physical properties, such as, vapor pressure, solubility parameter and viscosity [3, 4]. A MINLP model is established for the design of fragrance molecules. Decomposition-based solution approach is used to obtain the optimal result.

- **Data collection**

Create database. In this case study, the database developed by Keller et al. [80] is used. This database has 480 molecules. The molecules have between 1 and 28 nonhydrogen atoms, and, include 29 amines and 45 carboxylic acids. Two molecules contain halogen atoms, 53 have sulfur atoms, 73 have nitrogen atoms, and 420 have oxygen atoms. The molecules are structurally and chemically diverse, and many of them have unfamiliar smells, some have never been used in prior psychophysical experiments.

Select product descriptors. Fragment (group)-based representation is commonly used in GC methods and group-based QSPR methods. It has been shown that the properties of a molecule can be determined with relatively high accuracy by summation of the contributions of the associated groups. In this case study, 50 groups are selected as descriptors for ML modeling.

Collect product properties. Here, the odor pleasantness and odor characters are selected as the required key properties for a fragrance product, which are defined as follows. The odor pleasantness is a scale from the rating of people for a certain molecule, from 0 to 100; the odor characters are classified in terms of the following 20 categories based on people's perception [81], namely "edible," "bakery," "sweet," "fruit," "fish," "garlic," "spices," "cold," "sour," "burnt," "acid," "warm," "musky," "sweaty," "ammonia/urinous," "decayed," "wood," "grass," "flower," and "chemical." These 20 categories of odor characters cover most of the odors for the design of fragrance products in industry. To simplify the problem, only the key odor character is reserved for each molecule.

- **Data preprocessing and feature engineering**

As the input data (groups) have significant physical meaning without any data issue, data preprocessing, and feature engineering are not performed in this case study.

- **Model establishment**

With the obtained input (groups) and output data (odor pleasantness and odor characters), two convolutional neural networks (CNNs) models are established.

Model selection

CNNs are selected to correlate groups with the odor pleasantness and odor characters due to the following reasons:

- √ As is well-known, there is a unique parameter in CNN, called the kernel function. Since every input variable g_i is multiplied by a weight factor according to the convolutional calculation, the learning algorithm is capable of extracting features from the kernel. Thus, the output of CNN is also called a feature map. With this CNN character, essential groups, which map the odor of a molecule, can be selected by using the learning algorithm. Hence, the odor prediction models can be implemented without the disturbance of unimportant groups.
- √ Besides, the convolution calculation involves some insights, which could enhance the model performance, such as sparse interactions and parameter sharing. Sparse interactions mean the output is only affected by a few variables (groups), which could enlarge their influence on the odorant via the learning algorithm. Parameter sharing ensures the odorant prediction results are affected by all the molecular structural parameters as a whole.

Therefore, the above traits of CNN make it more suitable for the odor prediction than other neural network models.

Model training and validation process

Here, Python Keras [75] is used for the development of the CNN odor prediction models. The input to the model is a 50×1 vector of groups, and the output is odor properties, including odor characters and odor pleasantness. The layer information is shown in Fig. 7. In Keras, the embedding layers and the flatten layers are used for reshaping the data; the dropout layer is used to prevent overfitting; the dense layer is a fully connected layer, so the neurons in the layer are connected to those in the next layer. As shown in Fig. 7, the established CNN model structure consists a 50×64 embedding layer, a 47×128 convolutional layer, a 44×128 convolutional layer, a 22×128 max-pooling layer, a 22×128 dropout layer, a 2816×1 flatten layer, a 128×1 dense layer, another 128×1 dropout layer and 20×1 dense layer. Finally, the properties of odor characters and odor pleasantness are predicted using this model structure, with different trained parameters.

The 480 molecules in the database are trained using the established CNN models. The predicted results of odor characters and odor pleasantness are compared with the experimental data as shown in Figs. 8 and 9. It is not necessary to obtain continuous values of odor pleasantness, due to the odor properties are diverse among different people. Therefore, the odor pleasantness is discretized into 5 levels from the original odor pleasantness values (e.g., level 1: 0–20; level 2: 20–40, and so forth) to make the model more representative and applicability. Therefore, the prediction results shown in Fig. 8 are also discretized into five levels.

From the results in Figs. 8 and 9, it is seen that both the prediction of odor characters and pleasantness are accurate using the developed CNN models, which are trained using the 480 molecules in the database. The average correctness of odor characters is 92.9%, while the average prediction error of odor

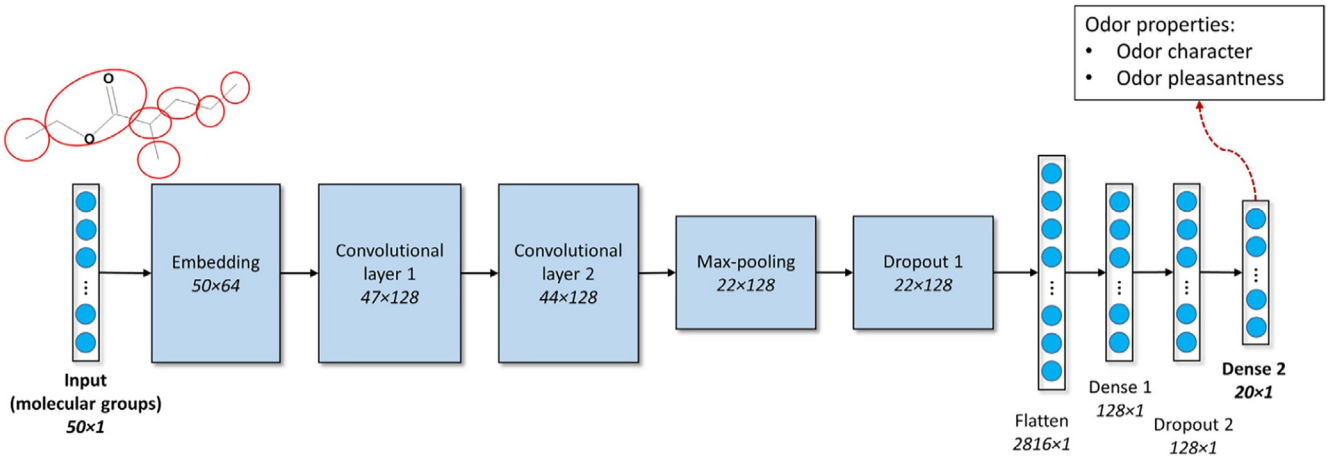


FIG. 7 CNN layer information for odor prediction model.

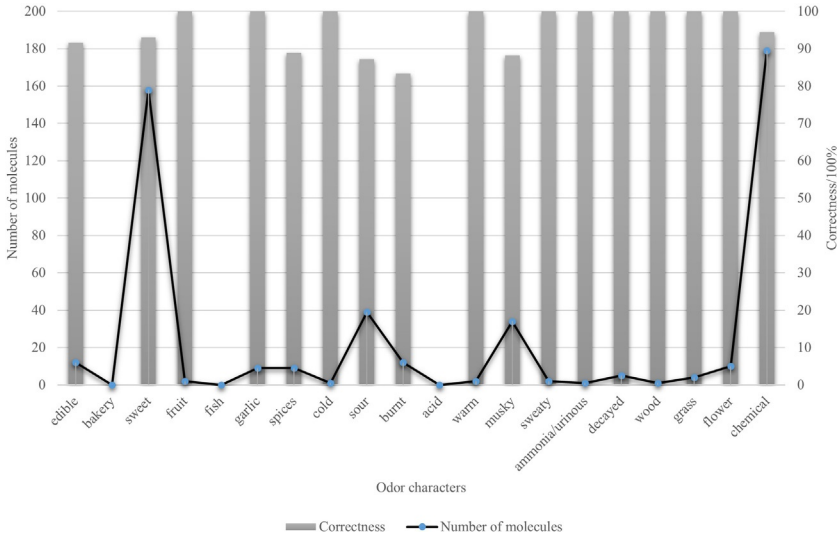


FIG. 8 Predicted results of 480 molecules in the database for odor characters.

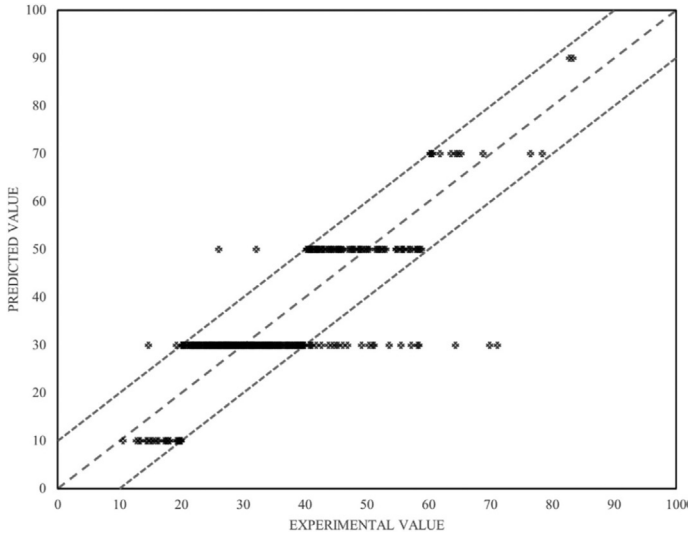


FIG. 9 Comparison of experimental values and predicted results of odor pleasantness (scale from 0 to 100) for the 480 molecules in the database.

pleasantness is 18.4%. In Fig. 8, the black line shows the number of molecules for a typical odor character in the database, while the bar shows the correctness of the model prediction. In the database, character "sweet" and "chemical" possess the largest amount of the molecules, while other ones possess smaller but sufficient numbers of molecules. In Fig. 9, the two dashed lines indicate the

TABLE 8 Predicted results of several commonly used fragrance molecules using the developed CNN models.

Molecule	Smells	Odor	Predicted odor character	Predicted odor pleasantness (scale from 0 to 100)	Correct?
Limonene	CC1=CCC(CC1)C(=C)C	A strong smell of oranges	Musky	20–40	N
Geraniol	OCC=C(C)CCC=C(C)C	Floral	Sweet	40–60	Y
Vanillin	O=Cc1ccc(O)c(OC)c1	Vanilla	Sweet	80–100	Y
Linalool	C=CC(O)(C)CCC=C(C)C	Sweet, floral, petitgrain-like	Sweet	40–60	Y

acceptable range for the predicted properties, that is, the predicted property (indicated by dots) must be inside region covered by the dashed lines. From Fig. 9, 27 molecules are out of the acceptable range. Since the odor pleasantness experimental values are obtained from the rating of people, the data may not be quite accurate. Therefore, although most of the characters and pleasantness have a satisfactory correctness, the prediction of odor characters for molecules outside the database has to be reevaluated. In Table 8, several fragrance molecules outside the database, which are commonly used in our daily life, are evaluated using the ML model. The evaluation results of Table 8 show roughly 75% correctness for molecules outside the database using the developed CNN models, which indicates that the trained CNN models are not overfitting.

- **Chemical product design**

The objective of this case study is to find suitable fragrance molecules as additives for shampoo, where the odor of the molecules is predicted using the developed CNN models while GC-based models are included to predict the rest of the needed physical properties, such as vapor pressure, solubility parameter and viscosity. The CAMD problem is formulated as a MINLP model for the design of fragrance molecules. The decomposition-based solution approach [70] is used

TABLE 9 Properties and constraints for fragrance molecule design.

Properties	Constraints
Total group number	$4 \leq n \leq 10$
Repeat group number	$n_i \leq 4$
Functional group number	$1 \leq n_F \leq 3$
Odor character	OC = sweet, fruit or flower
Odor pleasantness	$OP \geq 40$
Diffusion coefficient (m^2/h)	$D \geq 0.15$
Vapor pressure (Pa)	$P^{sat} \geq 100$
Normal Boiling point (K)	$T_b \geq 440$
Normal melting point (K)	$T_m \leq 293.15$
Solubility parameter ($MPa^{1/2}$)	$15 \leq S_p \leq 17$
Viscosity (cP)	$\eta \leq 2$
Density (g/cm^3)	$0.8 \leq \rho \leq 1$
$-\log(LC_{50})FM$ ($-\log$(mol/L))	$-\log(LC_{50})\,FM \leq 4.2$

to obtain the optimal result. The following groups are selected: CH_3, CH_2, CH, C, CH_2=CH, CH=CH, CH_2=C, CH=C, OH, CH_3CO, CH_2CO, CH_3COO, CH_2COO, CH_3O, CH_2O. The lower and upper bonds for structure and property constraints are given in Table 9. More detailed information can be found in our previous work [4].

First, feasible candidates are generated by matching constraints T_b, T_m, S_p, η, ρ and $-\log(LC_{50})FM$ as the model equations for these properties are linear. 40 Feasible molecules are generated in this subproblem using the OptCAMD software [82]. Then, constraints D and P_{sat} are added to evaluate each generated candidate to check if they satisfy these additional constraints. 26 Molecules are selected in this subproblem. Then, the 26 molecules are tested using the CNN model for odor character prediction, to test if these molecules are "sweet," "fruit," or "flower" (as defined in Table 9), and 8 molecules are found to match these constraints. The odor pleasantness CNN model is then used for the screening of these 8 molecules, which finds 6 molecules matching this constraint. The final solution is the molecule which has the highest odor pleasantness within these 6 molecules. The 6 generated molecules satisfying all property constraints are listed in Table 10, together with their properties.

TABLE 10 The generated feasible candidates.

No.	1	2	3	4	5	6
Formula	$C_8H_{16}O$	$C_8H_{16}O_2$	$C_7H_{12}O_2$	$C_7H_{12}O_3$	$C_8H_{14}O_2$	$C_9H_{18}O_2$
Groups	2 CH_3 4 CH_2 1 CH_2CO	1 CH_3 4 CH_2 1 CH_2CO 1 CH_3O	1 CH_3 1 CH 1 CH_2=C 1 CH_3CO 1 CH_3O	1 CH_3 1 CH 1 CH_2=CH 1 CH_3COO 1 CH_3O	1 CH_3 2 CH 1 CH_2=CH 1 CH_3CO 1 CH_3O	3 CH_3 3 CH_2 1 C 1 CH_3COO
T_m/K	244	265	253	217	253	240
T_b/K	443	469	443	442	459	458
S_p/Mpa$^{1/2}$	16.47	16.88	16.55	16.49	16.32	15.23
η/CP	1.08	0.91	0.21	0.2	0.15	0.89
ρ/g/cm^3	0.82	0.9	0.96	1	0.94	0.9
$-\log(LC_{50})$	3	2.58	2.83	3.53	3.58	3.13
P^{sat}/Pa	1003.8	138.2	838.4	1298.3	318.4	501.7
D/m^2/h	0.17	0.16	0.17	0.17	0.16	0.16
OC	Sweet	Sweet	Sweet	Sweet	Sweet	Sweet
OP	40	40	40	40	40	60
Available in database?	Y	Y	N	N	N	N
CAS number	111-13-7	106-73-0	–	–	–	–
Molecular structure			–	–	–	–
Odor in literature	Cheese-like, dairy nuances	Fruity	–	–	–	–

From the optimization result, molecule $C_9H_{18}O_2$ has the highest odor pleasantness. Therefore, it is selected as the best potential fragrance molecule in this case study. Database search has been performed for all the six feasible molecules. The optimal molecule, however, is not found in any database as fragrance and therefore, it needs to be evaluated through experiments to verify if the odor properties are the same as predicted. The molecules $C_8H_{16}O$ (CAS number: 111-13-7) and $C_8H_{16}O_2$ (CAS number: 106-73-0) are found in the database as commonly used fragrances for various purposes, which confirms the effectiveness of the fragrance molecule design model and its solution.

5 Conclusions

In this chapter, an optimization-based ML-CAMD framework is discussed in detail for the establishment of ML models for property prediction through model selection, training, and validation, as well as chemical product design through efficient mathematical optimization algorithms. Two case studies are

presented for the applications of the proposed framework, where HDNN and CNN models are respectively established for the predictions of the atomic surface charge density profiles and the odor pleasantness/characters. Afterwards, the ML models are successfully incorporated into the mixed-integer nonlinear programming models for computer-aided molecular design design. The model complexity is managed by a decomposition-based algorithm. Both case studies obtained the optimal products (crystallization solvents for case study 4.1 and fragrance molecules for case study 4.2) with satisfied performances, which will be further verified by experiments in our future work.

Acknowledgments

The authors are grateful for the financial support of National Natural Science Foundation of China (22078041, 21808025) and "the Fundamental Research Funds for the Central Universities (DUT20JC41)."

References

[1] L. Zhang, D.K. Babi, R. Gani, New vistas in chemical product and process design, Annu. Rev. Chem. Biomol. Eng. 7 (2016) 557–582.
[2] L. Zhang, H. Mao, Q. Liu, R. Gani, Chemical product design—recent advances and perspectives, Curr. Opin. Chem. Eng. 2020 (27) (2020) 22–34.
[3] L. Zhang, K.Y. Fung, C. Wibowo, R. Gani, Advances in chemical product design, Rev. Chem. Eng. 34 (3) (2018) 319–340.
[4] L. Zhang, H. Mao, L. Liu, J. Du, R. Gani, A machine learning based computer-aided molecular design/screening methodology for fragrance molecules, Comput. Chem. Eng. 115 (2018) 295–308.
[5] H. Cartwright, Development and Uses of Artificial Intelligence in Chemistry, Reviews in Computational Chemistry, John Wiley & Sons, Inc., New York, 2007, pp. 349–390.
[6] M.N.O. Sadiku, S.M. Musa, O.M. Musa, Machine learning in chemistry industry, Int. J. Adv. Sci. Res. Eng. 3 (10) (2017) 12–15.
[7] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[8] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.
[9] O. Ivanciuc, Applications of support vector machines in chemistry, Rev. Comput. Chem. 23 (2007) 291.
[10] H. Geppert, M. Vogt, J. Bajorath, Current trends in ligand-based virtual screening: molecular representations, data mining methods, new application areas, and performance evaluation, J. Chem. Inf. Model. 50 (2010) 205–216.
[11] P.R. Zonouz, A. Niaei, A. Tarjomannejad, Modeling and optimization of toluene oxidation over perovskite-type nanocatalysts using a hybrid artificial neural network-genetic algorithm method, J. Taiwan Inst. Chem. Eng. 65 (2016) 276–285.
[12] A. Velásco-Mejía, V. Vallejo-Becerra, A.U. Chávez-Ramírez, J. Torres-González, Y. Reyes-Vidal, F. Castañeda-Zaldivar, Modeling and optimization of a pharmaceutical crystallization process by using neural networks and genetic algorithms, Powder Technol. 292 (2016) 122–128.
[13] Y. Liu, T. Zhao, W. Ju, S. Shi, Materials discovery and design using machine learning, J. Mater. 3 (3) (2017) 159–177.

[14] P. Pankajakshan, S. Sanyal, O.E. de Noord, I. Bhattacharya, A. Bhattacharyya, U. Waghmare, Machine learning and statistical analysis for materials science: stability and transferability of fingerprint descriptors and chemical insights, Chem. Mater. 29 (10) (2017) 4190–4201.

[15] Q. Vanhaelen, A.M. Aliper, A. Zhavoronkov, A comparative review of computational methods for pathway perturbation analysis: dynamical and topological perspectives, Mol. BioSyst. 13 (1) (2017) 1692–1704.

[16] N. Hadi, A. Niaei, S.R. Nabavi, R. Alizadeh, M.N. Shirazi, B. Izadkhah, An intelligent approach to design and optimization of M-Mn/H-ZSM-5 (M: Ce, Cr, Fe, Ni) catalysts in conversion of methanol to propylene, J. Taiwan Inst. Chem. Eng. 59 (2016) 173–185.

[17] T. Zhou, S. Jhamb, X. Liang, K. Sundmacher, R. Gani, Prediction of acid dissociation constants of organic compounds using group contribution methods, Chem. Eng. Sci. 183 (2018) 95–105.

[18] W.L. Kubic Jr., R.W. Jenkins, C.M. Moore, T.A. Semelsberger, A.D. Sutton, Artificial neural network based group contribution method for estimating cetane and octane numbers of hydrocarbons and oxygenated organic compounds, Ind. Eng. Chem. Res. 56 (2017) 12236–12245.

[19] Y. Lo, S.E. Rensi, W. Torng, R.B. Altman, Machine learning in chemoinformatics and drug discovery, Drug Discov. Today 23 (8) (2018) 1538–1546.

[20] T. Le, V.C. Epa, F.R. Burden, D.A. Winkler, Quantitative structure–property relationship modeling of diverse materials properties, Chem. Rev. 112 (5) (2012) 2889–2919.

[21] Daylight Theory Manual, Chemical Information Systems, Inc., 2019. https://www.daylight.com/dayhtml/doc/theory/theory.finger.html. (Accessed 10 August 2019).

[22] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inf. Model. 50 (5) (2010) 742–754.

[23] J.N. Wei, D. Duvenaud, A. Aspuru-Guzik, Neural networks for the prediction of organic chemistry reactions, ACS Cent. Sci. 2 (10) (2016) 725–732.

[24] H.S. Stein, D. Guevarra, P.F. Newhouse, E. Soedarmadji, J.M. Gregoire, Machine learning of optical properties of materials—predicting spectra from images and images from spectra, Chem. Sci. 10 (2019) 47–55.

[25] B. Meyer, B. Sawatlon, S. Heinen, O.A. von Lilienfeld, C. Corminboeuf, Machine learning meets volcano plots: computational discovery of cross-coupling catalysts, Chem. Sci. 9 (35) (2018) 7069–7077.

[26] F. Häse, I.F. Galván, A. Aspuru-Guzik, R. Lindhb, M. Vacher, How machine learning can assist the interpretation of ab initio molecular dynamics simulations and conceptual understanding of chemistry, Chem. Sci. 10 (2019) 2298–2307.

[27] M.A. Kayala, C. Azencott, J.H. Chen, P. Baldi, Learning to predict chemical reactions, J. Chem. Inf. Model. 51 (9) (2011) 2209–2222.

[28] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional Networks on Graphs for Learning Molecular Fingerprints, 2015. arXiv preprint. arXiv:1509.09292.

[29] S. Kearnes, K. Mccloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Comput. Aided Mol. Des. 30 (8) (2016) 595–608.

[30] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Model. 28 (1988) 31–36.

[31] M.H. Segler, T. Kogej, C. Tyrchan, M.P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks, ACS Cent. Sci. 4 (1) (2018) 120–131.

[32] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[33] M.S. Nixon, A.S. Aguado, Feature Extraction & Image Processing for Computer Vision, Academic Press, 2012.

[34] G. Chaudhary, S. Srivastava, S. Bhardwaj, Feature extraction methods for speaker recognition: a review, Int. J. Pattern Recognit. Artif. Intell. 31 (12) (2017) 1750041.
[35] I. Levner, V. Bulitko, G. Lin, Feature extraction for classification of proteomic mass spectra: a comparative study, in: Feature Extraction, Springer, Berlin, Heidelberg, 2006, pp. 607–624.
[36] A.R. Tate, D. Watson, S. Eglen, T.N. Arvanitis, E.L. Thomas, J.D. Bell, Automated feature extraction for the classification of human in vivo 13C NMR spectra using statistical pattern recognition and wavelets, Magn. Reson. Med. Off. J. Soc. Magn. Reson. Med. 35 (6) (2010) 834–840.
[37] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[38] H.D. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, 9, Morgan Kaufmann, San Mateo, 1997, pp. 155–161.
[39] B. Schölkopf, A.J. Smola, Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, 2002.
[40] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[41] Y. Zhao, X. Zhang, L. Deng, S. Zhang, Prediction of viscosity of imidazolium-based ionic liquids using MLP and SVM algorithms, Comput. Chem. Eng. 92 (2016) 37–42.
[42] F. Luan, H.T. Liu, Y.Y. Wen, X.Y. Zhang, Classification of the fragrance properties of chemical compounds based on support vector machine and linear discriminant analysis, Flavour Fragr. J. 23 (4) (2008) 232–238.
[43] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[44] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Chapman & Hall/CRC, Boca Raton, FL, 1984.
[45] R.J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[46] S.K. Murthy, S. Kasif, S. Salzberg, A system for induction of oblique decision trees, J. Artif. Intell. Res. 2 (1994) 1–32.
[47] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[48] S. Datta, V.A. Dev, M.R. Eden, Hybrid genetic algorithm-decision tree approach for rate constant prediction using structures of reactants and solvent for Diels-Alder reaction, Comput. Chem. Eng. 106 (2017) 690–698.
[49] B. Partopour, R.C. Paffenroth, A.G. Dixon, Random forests for mapping and analysis of microkinetics models, Comput. Chem. Eng. (2018).
[50] D.S. Cao, Y.N. Yang, J.C. Zhao, J. Yan, S. Liu, Q.N. Hu, Y.Z. Liang, Computer-aided prediction of toxicity with substructure pattern and random forest, J. Chemom. 26 (1–2) (2012) 7–15.
[51] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Syst. 2 (3) (1988) 321–355.
[52] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by backpropagating errors, Nature 323 (1986) 533–536.
[53] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: 2004 IEEE International Joint Conference on Neural Networks, 2004. Proceedings, vol. 2, IEEE, 2004, pp. 985–990.
[54] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[55] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[56] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, in: M.A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, 1995.

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, Springer, Berlin, 2014, pp. 2672–2680.

[58] H. Schwenk, Y. Bengio, Training methods for adaptive boosting of neural networks, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 1998, pp. 647–653.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[60] G.E. Hinton, Tutorial on Deep Learning, IPAM Graduate Summer School: Deep Learning, Feature Learning, 2012.

[61] D. Kingma, J.B.J.C. Science, Adam: A Method for Stochastic Optimization, 2014. arXiv e-prints. arXiv:1412.6980.

[62] A.Z. Hezave, M. Lashkarbolooki, S. Raeissi, Using artificial neural network to predict the ternary electrical conductivity of ionic liquid systems, Fluid Phase Equilib. 314 (2012) 128–133.

[63] P. Díaz-Rodríguez, J.C. Cancilla, G. Matute, J.S. Torrecilla, Viscosity estimation of binary mixtures of ionic liquids through a multi-layer perceptron model, J. Ind. Eng. Chem. 21 (2015) 1350–1353.

[64] M. Spellings, S.C. Glotzer, Machine learning for crystal identification and discovery, AICHE J. 64 (6) (2018) 2198–2206.

[65] T. Gao, J.R. Kitchin, Modeling palladium surfaces with density functional theory, neural networks and molecular dynamics, Catal. Today 312 (2018) 132–140.

[66] J.R. Boes, J.R. Kitchin, Neural network predictions of oxygen interactions on a dynamic Pd surface, Mol. Simul. 43 (5–6) (2017) 346–354.

[67] A. Ethem, Design and analysis of machine learning experiments, in: T. Dietterich (Ed.), Introduction of Machine Learning, second ed., The MIT Press, Cambridge, Massachusetts London, England, 2009, pp. 475–514.

[68] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors, 2012. arXiv e-prints. arXiv:1207.0580.

[69] A.Y. Ng, Feature selection, L1 vs. L2 regularization, and rotational invariance, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, p. 78.

[70] A.T. Karunanithi, L.E. Achenie, R. Gani, A new decomposition-based computer-aided molecular/mixture design methodology for the design of optimal solvents and solvent mixtures, Ind. Eng. Chem. Res. 44 (13) (2005) 4785–4797.

[71] E. Mullins, R. Oldland, Y.A. Liu, et al., Sigma-profile database for using COSMO-based thermodynamic methods, Ind. Eng. Chem. Res. 45 (12) (2006) 4389–4415.

[72] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, P. Marquetand, wACSF—weighted atom-centered symmetry functions as descriptors in machine learning potentials, J. Chem. Phys. 148 (24) (2018) 241709.

[73] C.-M. Hsieh, S.I. Sandler, S.-T. Lin, Improvements of COSMO-SAC for vapor–liquid and liquid–liquid equilibrium predictions, Fluid Phase Equilib. 297 (1) (2010) 90–97.

[74] G. Xavier, B. Antoine, B. Yoshua, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics: PMLR, 2011, pp. 315–323.

[75] F. Chollet, et al., Keras: The Python Deep Learning Library, Astrophysics Source Code Library, 2018. ascl:1806.1022.
[76] T.E. Oliphant, Python for scientific computing, Comput. Sci. Eng. 9 (3) (2007) 10–20.
[77] Y. Zhao, J. Chen, Q. Liu, Y. Li, Profiling the structural determinants of aryl benzamide derivatives as negative allosteric modulators of mGluR5 by in Silico study, Molecules 25 (2) (2020).
[78] A.S. Hukkerikar, B. Sarup, A. Ten Kate, J. Abildskov, G. Sin, R. Gani, Group-contribution+ (GC+) based estimation of properties of pure components: improved property estimation and uncertainty analysis, Fluid Phase Equilib. 321 (2012) 25–43.
[79] A.T. Karunanithi, L.E. Achenie, R. Gani, A computer-aided molecular design framework for crystallization solvent design, Chem. Eng. Sci. 61 (4) (2006) 1247–1260.
[80] A. Keller, R.C. Gerkin, Y. Guan, A. Dhurandhar, G. Turu, B. Szalai, J.D. Mainland, Y. Ihara, C.W. Yu, R. Wolfinger, Predicting human olfactory perception from chemical features of odor molecules, Science 355 (2017) 820.
[81] A. Keller, L.B. Vosshall, Olfactory perception of chemically diverse molecules, BMC Neurosci. 17 (2016) 55.
[82] Q. Liu, L. Zhang, L. Liu, J. Du, A.K. Tula, M. Eden, R. Gani, OptCAMD: an optimization-based framework and tool for molecular and mixture product design, Comput. Chem. Eng. 124 (2019) 285–301.
[83] Q. Liu, L. Zhang, K. Tang, L. Liu, J. Du, Q. Meng, R. Gani, Machine learning-based atom contribution method for the prediction of surface charge density profiles and solvent design, AIChE J. 67 (2) (2021) e17110.