https://doi.org/10.1016/j.ces.2021.117219

Insights into Ensemble Learning-based Data-Driven Model for Safety-Related Property of Chemical Substances

Zihao Wang [1], Huaqiang Wen [1], Yang Su [2], Weifeng Shen [1,*], Jingzheng Ren [3], Yingjie Ma [4], Jie Li [4]

[1] *School of Chemistry and Chemical Engineering, Chongqing University, Chongqing 400044, China*

[2] *School of Intelligent Technology and Engineering, Chongqing University of Science and Technology, Chongqing, 401331, China*

[3] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China*

[4] *Centre for Process Integration, Department of Chemical Engineering and Analytical Science, The University of Manchester, Manchester M139PL, U.K.*

Correspondence should be addressed to **Weifeng Shen at shenweifeng@cqu.edu.cn*

Abstract

Risk assessment relying on characteristics of chemicals in process industries can prevent accidents caused by flammable and combustible liquids and gases. Whereas its application is limited by the lack of safety-related properties for abundant chemicals of interest, which promotes the demand for accurate predictive models to evaluate inherent safety implications of chemicals. In this research, staking-based ensemble learning is comprehensively investigated on safety-related properties to assist the risk assessment. Based on molecular structure-based features, individual and ensemble models are built and compared using heterogeneous machine learning (ML) methods. The systematic ensemble learning workflow is deployed by a case on flash points of chemical substances. Several representative ML methods including multiple linear regression, extreme learning machine, feedforward neural network, and support vector machine are taken into consideration. As it turns out, ensemble models exhibit improved predictive accuracy than standard individual ML models, indicating the effectiveness of ensemble learning on improving model performance. Moreover, extremal

evaluations with existing models as well as internal analyses against functional group-based organic compound families and structural feature-based data-driven categories are carried out to identify model reliability. Ensemble learning is demonstrated as an effective approach for high-performance predictive modeling in safety-related risk assessments.

Keywords: machine learning; predictive modeling; molecular feature; flash point

1 Introduction

Safety-related properties of organic compounds play a crucial role and have an extensive application in chemical and environmental engineering[1]. Flash point, auto-ignition temperature, flammability limits, and other properties are essential in performing risk assessments. Although experimental measurements are fairly accurate and reliable, they are expensive, time-consuming, and sometimes dangerous. Therefore, computer-aided property prediction is recognized as an alternative solution to handle these problems, and from the perspective of process safety and risk management, it accelerates computer-aided product design for the safe operation of industrial processes.

Over the past decades, a wide variety of advanced computational approaches, such as group contribution (GC) methods[2,3] and traditional machine learning (ML) algorithms[4-6] have been put forward and broadly applied to develop predictive models to accurately calculate properties of chemical substances. Most recently, cutting-edge deep learning techniques[7-9] also have been creatively and successfully implemented with the same objectives. Accurate predictive models drive the computer-aided molecular and product design toward high efficiency[10-12], which are important to accelerate marketing by focusing on the best candidates at the earliest research and development stage[13-15].

Linear regression is one of the most rudimentary and explainable approaches in mathematical modeling, characterized by simple structure and quick calculation[16]. The traditional GC methods, especially the popular three-level GC methods[17], were proposed coupling multiple linear regression (MLR) for property predictive modeling[18-20]. Differing

from the MLR, a feedforward neural network (FNN) can handle complicated tasks by introducing non-linear transformations. Thus, FNN models usually make more accurate and reliable predictions. Attributed to its advantages in data mining, the FNN has been extensively deployed in property prediction using different molecular descriptors[21-25]. Extreme learning machine (ELM) is a simplified form of FNN with a part of parameters are randomly predefined. Therefore, ELM has fewer learnable parameters and it is trained much faster than FNN[26]. Another popular ML algorithm, support vector machine (SVM) can efficiently perform complex non-linear tasks (particularly for high dimensional inputs) by introducing different kernel functions[27]. The SVM modeling presents excellent performance for its convex optimization, and it is therefore considered promising in property prediction[28,29].

Significantly increased computational power allows researchers to implement large-scale predictive models within a reasonable time. In the context of desiring higher-level requirements (mostly regarding model accuracy) on prediction tasks, a large number of innovative algorithms are proposed targeting the most challenging and popular applications, which have large-scale architectures and require huge computational resources. Developing advanced algorithms needs knowledge and experience from computer science, and thus it is pretty challenging and impractical for ML practitioners instead of ML experts and developers.

Ensemble learning is dedicated to predictive modeling by integrating simple ML models to achieve better performance and higher computational efficiency. It circumvents complex advanced ML models[30-32], and it is thereby more friendly to ML practitioners. For instance, random forest aggregates result from numerous decision trees to comprehensively make final predictions. As a bagging algorithm, random subsets of samples, as well as features, are used to train individual trees (i.e., homogeneous learners), requiring numerous trees to fully take all samples and features into account. In comparison, the stacking-based ensemble method is different. It develops several independent models in parallel using different approaches (i.e., heterogeneous learners), and on this basis, a meta-model is trained to implement the ensemble for final predictions. In contrast with the bagging method, the stacking method uses all features and samples for model development, and thus fewer models and computational effort are

required for ensemble learning, as presented in some recent works about toxicity and carcinogenicity research[33,34].

To efficiently build models using staking-based ensemble learning, utilizing computational resources reasonably for model improvements could be of great importance for ML practitioners to obtain desired models straightforwardly. In addition, discovering characteristics of favorable models in ensemble learning can also accelerate the implementation of ensembles. Therefore, in this work, staking ensembles are comprehensively investigated and compared using representative heterogeneous ML algorithms including MLR, ELM, FNN, and SVM. Individual models are trained based on independent ML methods, and on this basis, ensemble models are developed using an explainable weight-based linear combination. To gain insight into ensemble implementation, individual and ensemble models are compared elaborately to analyze the role of the independent ML model in staking ensembles via the deployment on the flash point case. In this way, key factors that influence ensemble performance are disclosed for the high-efficient implementation of ensemble learning in predictive modeling. Moreover, model evaluations and analyses are carried out externally with existing models and performed internally regarding functional-group-based organic compound families as well as structural-feature-based data-driven categories to exhibit model advantages.

2 Predictive modeling methodology

The stacking-based ensemble learning workflow is mainly deployed with five stages as illustrated in **Figure 1**, including the first three stages for data processing and the last two stages for model development.

- Dataset construction: collect molecular structures and property data to form a dataset;
- Data splitting: divide the dataset into subsets for model training, validation, and predictability evaluation;
- Feature extraction: generate feature vectors to characterize molecular structures;
- Individual modeling: develop individual models using heterogeneous ML algorithms;
- Ensemble modeling: develop ensemble models by integrating individual models.

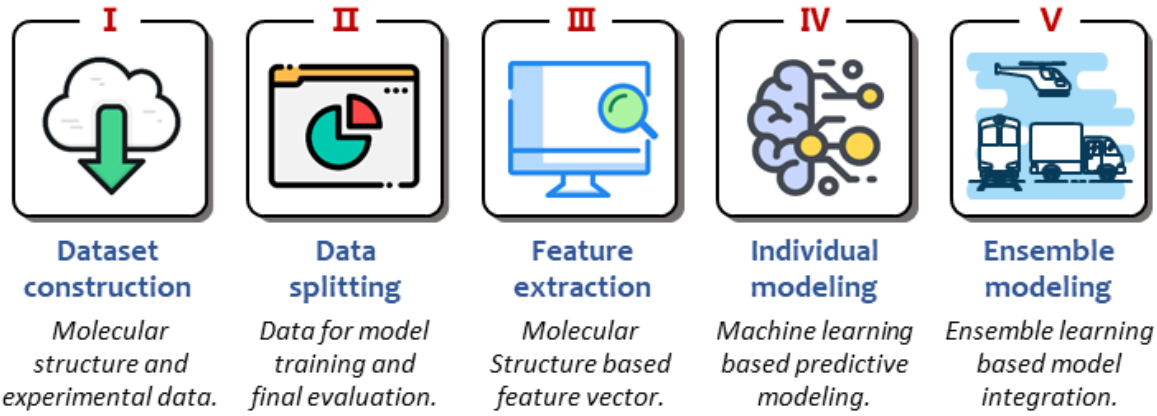

Figure 1. The workflow for implementing model development and ensemble.

2.1 Dataset construction

Molecular structural information and experimental targets are key components for predictive modeling. A language-like description, SMILES (Simplified Molecular Input Line Entry Specification) string can conveniently express molecular two- and three-dimensional structural characteristics by utilizing cheminformatics tools, and it is thus employed to characterize molecular structures. Experimental data should be collected from databases or literature to guarantee the reasonability of predictive models. As the flash point is employed as a case study for the deployment of ensemble-based predictive modeling, experimental flash point values (reported in the unit of K) and SMILES strings of organic compounds are gathered from the DIPPR 801 database[35] and PubChem[36]. Inorganic compounds (e.g., carbon monoxide, diborane, hydrogen sulfide) and metal-organic compounds (i.e., the organic compounds containing metal atoms such as sodium, mercury, or/and aluminum) are excluded from the gathered dataset. Therefore, the flash point dataset consisting of 1732 organic compounds covers various molecular structures including aliphatic and aromatic hydrocarbons, alcohols and phenols, heterocyclic compounds, amines, acids, ketones, esters, aldehydes, ethers, and organic halogen compounds, which indicates its chemical diversity for model development. The analysis for families of organic compounds is presented in **Table 1**, and the distribution of experimental values is provided in **Figure S1**.

Table 1. Detailed analysis for families of organic compounds in the employed dataset.

Family	Whole dataset
Aliphatic and aromatic hydrocarbons	465
Alcohols and phenols	177
Heterocyclic compounds	75
Amines	146
Carboxylic acids	94
Ketones	45
Esters	213
Aldehydes	38
Ethers	30
Organic halogen compounds	155
Others	294
Total	***1732***

2.2 Data splitting

Before training ML-based predictive models, the dataset needs to be divided into two disjoint subsets: a training set for model configuration and optimization, and a test set for evaluating the external predictive performance of the developed model, respectively accounting for 80% and 20% of the dataset. Parameters in ML models (e.g., weights and biases of neurons in FNN) can be learned from the training data, whereas the hyper-parameters should be customized by optimization.

Cross-validation is a model validation technique for assessing how well a model generalizes to new data. In *k*-fold cross-validation, the training set is equally partitioned into *k* subsets and the training routine is carried out *k* times. During each training process, one out of *k* subsets is assigned for model validation and the remaining *k*-1 subsets are used for model development. Therefore, each subset is used for model development *k*-1 times and model validation once. Finally, the performance of model configuration is evaluated with *k* independent validation sets in the *k*-fold cross-validation. In this way, limited data are fully devoted to discover a more robust model configuration for subsequent predictive modeling.

Herein, *k* is set to five, and as such, the five-fold cross-validation (as illustrated in **Figure2**) is employed. Hyper-parameters of ML models are determined by the grid search method.

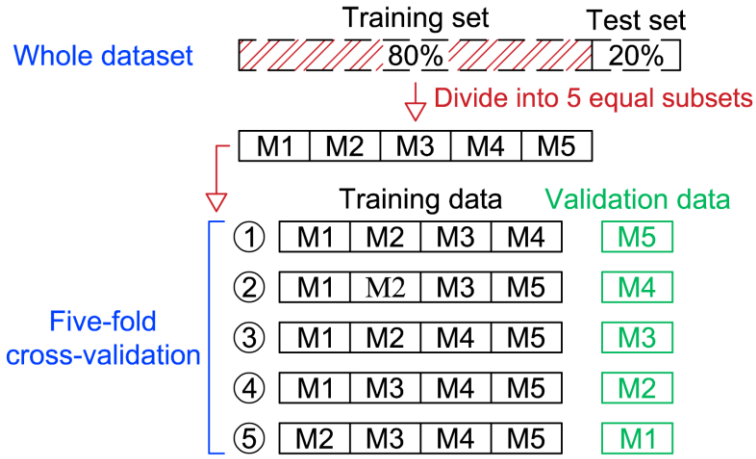

Figure 2. Schematic diagram of data splitting in the ML-based model development.

2.3 Feature extraction

To enable the ML model to read molecular structure, informative molecular features should be extracted in the form of numerical vector with a fixed length serving as input variables. For this, a novel feature extraction strategy is employed to automatically recognize molecular structures and extract molecular features in this research, which has been proven promising in generating molecular structure related descriptors for predictive modeling[37]. Each molecular feature represent a molecular substructure that only contains single non-hydrogen atom (accompanied with its connected hydrogen atoms and chemical bonds). Additionally, chemical information (i.e., the type and formal charge of the non-hydrogen atom, number of hydrogen atoms, types of bonds between substructures and stereo-centers) is fully considered in molecular features. For a specific molecule, occurrences of these extracted molecular features form a fixed-length numerical vector to characterize this molecule in ML models. The feature extraction and vector generation are simply exemplified in **Figure S2**.

Molecular structure represented with the SMILES strings are provided to perform the feature extraction. Relying on feature vectors, molecular structures are processed by ML algorithms, finally generating predicted values. Therefore, the feature extraction process is integrated with ML algorithm to achieve predictive modeling, illustrated in **Figure 3** using the

FNN framework as a ML instance.

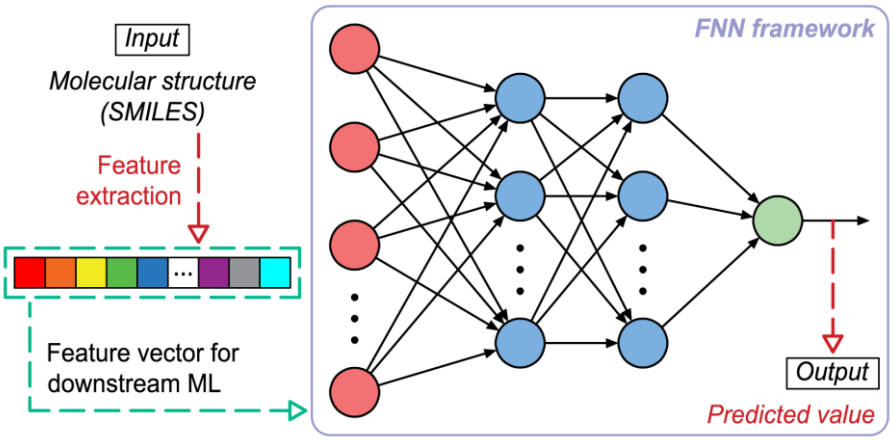

Figure 3. Integration of feature extraction process and ML algorithm for predictive modeling.

2.4 Individual modeling

Individual predictive models are developed using popular heterogeneous ML algorithms including MLR, ELM, FNN, and SVM. MLR is the simplest algorithm to model the linear relationship and it can easily achieve generality and interpretability, although it has a limitation on capturing the complex relationship. As indicated in Formula (1), MLR algorithm fits the target value relying on independent variables. Parameters in the multiple linear model are optimized by minimizing the objective function (i.e., sum of squares of residuals presented in Formula (2)) of all samples used for regression.

$$f(c) = \sum_{i=1}^{n}(\beta_i x_i) + \beta_0 \quad (1)$$

$$min \sum_{i=1}^{n}(f(x_i) - y_i)^2 \quad (2)$$

where β_0 is the constant term; β_i is the weight for each independent variable x_i; $\boldsymbol{x}$ is the multivariate matrix; n is the number of variables; y and $f(\boldsymbol{x})$ are the experimental and predicted values, respectively; N is the number of samples.

In comparison, FNN is the most widely used method in modeling complex relationship by introducing non-linear transformations. In the FNN architecture, input information is transferred from the first layer to the last layer passing through neurons and connections, and the mathematical transformation in neurons is illustrated with Formula (3). As indicated in Formulas (4)-(7), back-propagation (BP) and Adam[38] algorithms are employed to update

parameter θ and train the ML model *via* minimizing the objective function presented in Formula (8). Momentum, which accumulates the past gradients to determine the optimization direction, is taken into consideration in the Adam optimizer to update parameters, trying to escape the local minimum.

$$f(\boldsymbol{x}) = F(\sum_{i=1}^{n}(w_i x_i) + b) \tag{3}$$

$$v_t = \beta_1 \cdot v_{t-1} - (1-\beta_1)\cdot g_t \tag{4}$$

$$s_t = \beta_2 \cdot s_{t-1} - (1-\beta_2)\cdot {g_t}^2 \tag{5}$$

$$\Delta\theta_t = -\frac{v_t}{\sqrt{s_t+\varepsilon'}}\cdot g_t \tag{6}$$

$$\theta_{t+1} = \theta_t + \eta\cdot\Delta\theta_t \tag{7}$$

$$\sqrt{\frac{\sum_{i=1}^{N}(f(x_i)-y_i)^2}{N}} \tag{8}$$

where w and b are the weight and bias of neuron; g_t, v_t, and s_t are gradient, exponential average of gradients and exponential average of squares of gradients at time t; β_1, β_2 and ε' are parameters in Adam algorithm, and they are 0.9, 0.999 and 1.00×10^{-8}; η is the learning rate, and it is 2.00×10^{-3}; F is activation function (explained below).

The FNN model is constructed with two hidden layers in this work, and activation functions sigmoid and softplus, represented with Formulas (9) and (10), are assigned to hidden layers aiming at introducing non-linear transformation into neurons to enhance the ability of the FNN model in data fitting.

$$F(x) = \frac{1}{1+e^{-x}} \tag{9}$$

$$F(x) = ln\,(1+e^{-x}) \tag{10}$$

ELM has a similar structure to the FNN, while part of its parameters is assigned randomly before training. Thus, comparing to FNN, ELM is less accurate but it offers advantages in efficiency and generalization. Due to its less learnable parameters, the ELM model is thereby trained much faster than traditional FNN models. Activation function sigmoid is employed for the hidden layer of the ELM model, as presented in Formula (9).

In addition, SVM is also a popular method that is especially preferred to high-dimensional data, and it can efficiently handle non-linear data using the kernel trick. Another individual

model is developed with the SVM algorithm as indicated in Formula (11), and hyperplanes are created to fit data points. In order to improve SVM models in handling complex regression tasks, the Gaussian radial basis function, as presented in Formula (12), is employed as the kernel method. The SVM model is optimized *via* minimizing the objective function, Formula (13), ensuring that all data points are as close to the created hyperplanes as possible.

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)G(x_i, x) + a \quad (11)$$

$$G(x_i, x_j) = exp\,(-\gamma\|x_i - x_j\|^2) \quad (12)$$

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i, x_j) + \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*) \quad (13)$$

Formula (13) is subjected to: $\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$

$$\forall i: 0 \le \alpha_i \le C$$

$$\forall i: 0 \le \alpha_i^* \le C$$

where α and α^* are Lagrange multipliers; a is the distance term; $G(x_i,x_j)$ represents the kernel function; γ is the coefficient in Gaussian radial basis function; ε is the tolerance (i.e., acceptable error margin); C is the regularization coefficient.

2.5 Ensemble modeling

The identical training set in individual modeling is used for the subsequent ensemble learning. For this, a linear weight-based combination is applied to individual ML models for ensemble purposes. In the ensemble model, each individual predictor holds weight as indicated in Formula (14).

$$f(z) = \sum_{n=1}^{N}(k_i z_i) + c \quad (14)$$

where k_i is the weight of individual predictor; z_i is the predicted value from the individual predictor; c is the error term. Ensemble models are optimized by minimizing the sum of squares of residuals between ensemble results and target values.

3 Results and discussion

3.1 Development of individual models

In terms of theMLR model, its model structure is specified by the independent variables (i.e., molecular features summarized in **Table S1**). Therefore, weights of variables and the error

term in the linear model are determined by minimizing the sum of squares of residuals to achieve better predictive accuracy. Thus, the MLR model can be obtained by fitting on the training set. Weights of molecular features and the error term of the MLR model are provided in **Table S3**. Regarding the ELM model, hyper-parameters (i.e., number of units in the hidden layer and mixing coefficient of activation) are optimized with the five-fold cross-validation. As shown in **Figure S3**, the optimal ELM configuration has 270 units and a mixing coefficient of 0.81, which is employed for the subsequent model development.

Concerning the FNN model, parameters including weights and biases are learned from the training data and updated using the Adam optimizer, which presents advantages in computational efficiency. To construct an FNN model with strong robustness, five-fold cross-validation is used to optimize the FNN structure (i.e., numbers of neurons in the hidden layers). In the five-fold cross-validation, the model structure showing the lowest average loss is recognized as the optimal one for subsequent modeling. As shown in **Figure 4**, the FNN structure consisting of 11 and 6 neurons in two hidden layers exhibits the lowest RMSE value, and it is therefore used for the construction of the FNN model.

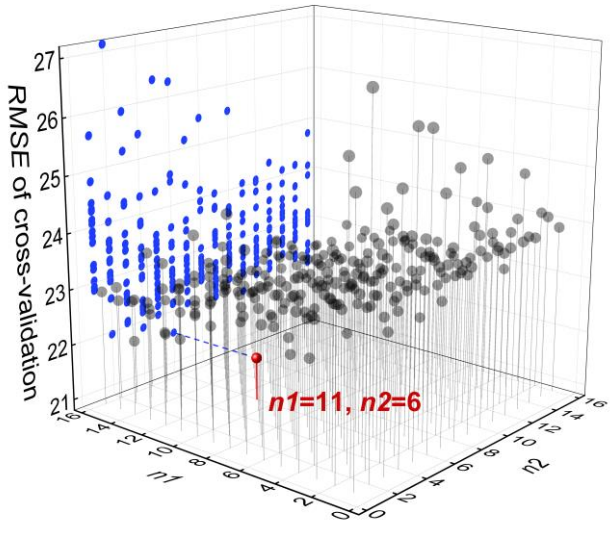

Figure 4. Cross-validation-based optimization for hyper-parameters (numbers of neurons in two hidden layers, n1 and n2) of the FNN model using the grid search.

The predictive performance of SVM models depends heavily on their hyper-parameters. Therefore, important hyper-parameters (i.e., regularization parameter C and tolerance ε) in the

SVM algorithm should be carefully chosen to improve the model performance. Thus, five-fold cross-validation is used to seek the appropriate combination of these parameters. As it turns out, in the case of that C and ε equal to 2340 and 5, the SVM model presents the lowest RMSE value (see **Figure S4**). Therefore, these optimal hyper-parameters are used for developing the SVM model.

At this point, optimal model structures have been obtained, and on this basis, individual predictive models are determined with the training set using the MLR, ELM, FNN, and SVM algorithms. Model parameters of four predictive models are summarized in **Tables S3-S6**. The test set is employed as a new dataset to validate the external performance of the developed predictive models. Therefore, statistical metrics including root mean square error (RMSE) and determination coefficient (R^2) are calculated to quantitatively evaluate these models as summarized in **Table 2**.

Table 2. Statistical metrics of MLR, ELM, FNN, and SVM models on the test set.

Model	N	RMSE (K)	R^2
MLR model	346	32.7255	0.8118
ELM model	346	20.5099	0.9261
FNN model	346	17.0513	0.9489
SVM model	346	15.9377	0.9554

Note: N is the number of data points; lower RMSE and higher R^2 are preferred.

It should be noted that model development and evaluation are carried out on identical training and test data to guarantee fair comparisons. Evaluated by the test set, the SVM model presents good predictive accuracy with an R^2 of 0.9554, which shows better performance than other models (i.e., the FNN model, followed by ELM and MLR models). Moreover, their predictive performance is visualized with the parity plots as shown in **Figure 5**. As it can be seen, there are a couple of data points present huge prediction deviations for the MLR model, which is caused by its simple linear structure and limited ability in handling modeling tasks. Despite that MLR, ELM, and FNN models exhibit relatively inaccurate predictions in this case, they learned underlying relationships between molecular structures and properties. Therefore,

they are also able to play a role in collaborative forecasting with less significant contributions, which is discussed in Section 3.2 by performing ensemble learning on these individual models.

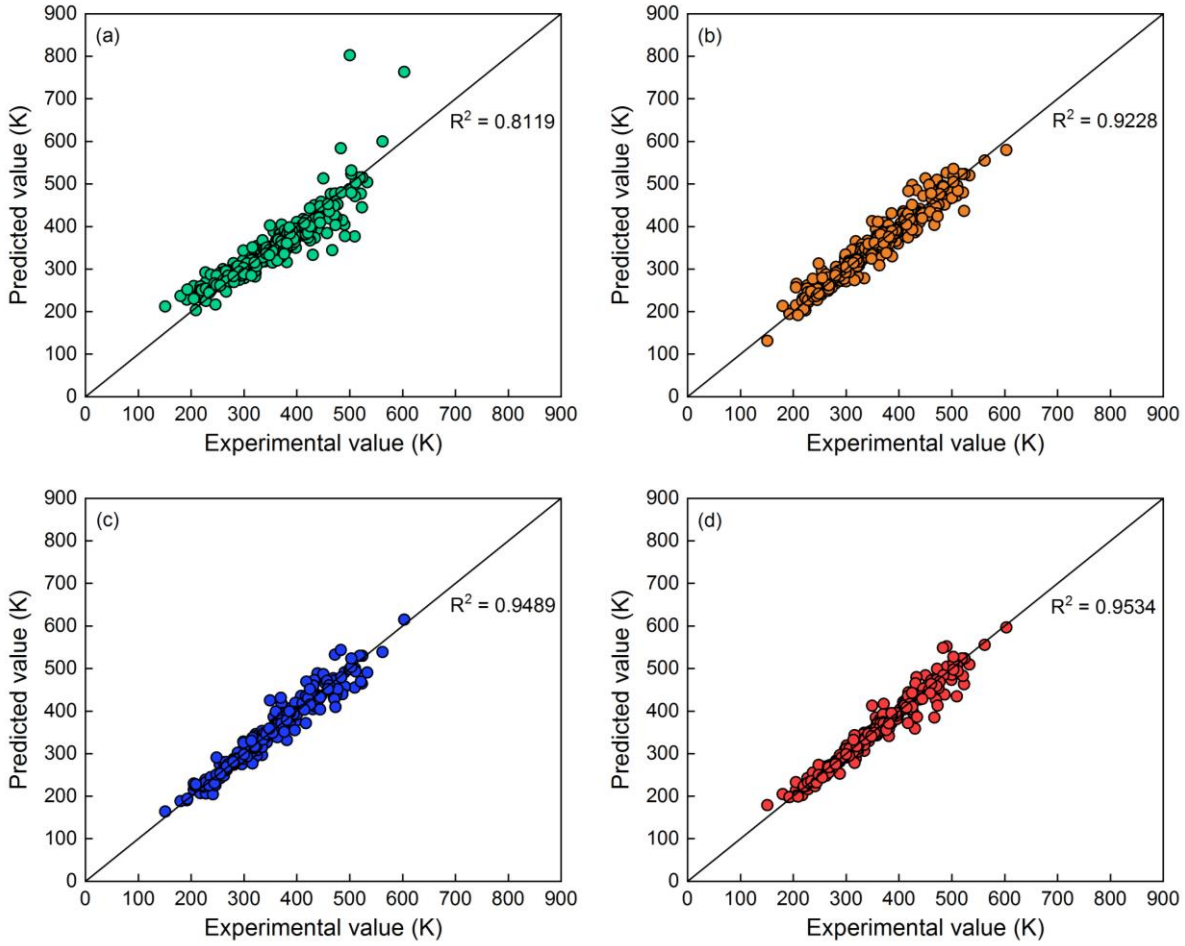

Figure 5. Parity plots for flash point predictions on the test set: (a) MLR model; (b) ELM model; (c) FNN model; (d) SVM model.

3.2 Development of ensemble models

To develop ensemble models, the most straightforward linear method is used to determined weights of individual heterogeneous models, which can display and analyze the contribution of each model. Relying on the MLR, ELM, FNN, and SVM individual models, a total of 11 ensemble models are developed in terms of different combinations (i.e., 6 two-predictor, 4 three-predictor, and 1 four-predictor ensemble models). Likewise, ensemble models are optimized by minimizing the sum of squares of residuals to fix the weights of individual models.

Statistical analyses are performed to quantify the predictive accuracy of ensemble models

on the test set, as summarized in **Table 3**. The four-predictor model shows a lower RMSE value and a higher R^2 value, indicating its better predictive accuracy. In general, the four-predictor model is slightly better than the three-predictor models and is much better than the two-predictor models. It seems that the predictive accuracy of the ensemble model can potentially be improved by enriching the diversity of heterogeneous learning algorithms. The improved accuracy indicates the effectiveness of ensemble learning in predictive modeling.

Table 3. Statistical metrics of ensemble predictive models on the test set.

Model type	Model	RMSE (K)	R^2
Two-predictor	MLR-ELM model	19.2828	0.9347
	MLR-FNN model	16.9795	0.9493
	ELM-FNN model	16.4765	0.9523
	MLR-SVM model	16.0198	0.9549
	ELM-SVM model	15.7641	0.9563
	FNN-SVM model	15.4381	0.9581
Three-predictor	MLR-ELM-FNN model	16.4389	0.9525
	MLR-ELM-SVM model	15.8653	0.9558
	MLR-FNN-SVM model	15.4684	0.9580
	ELM-FNN-SVM model	15.4194	0.9582
Four-predictor	MLR-ELM-FNN-SVM model	15.4463	0.9581

It is noted that, for example, the MLR-ELM-FNN model (RMSE of 16.4389) performs worse than the FNN-SVM model (RMSE of 15.4381), although it integrates more learning algorithms. In this case, the SVM model (RMSE of 15.9377) is better than the FNN model 17.0513). It is supposed that ensemble modeling is dominated by the best individual component. Thus, the FNN-SVM model shows better performance, since SVM controls ensemble modeling with its higher accuracy. To build better ensemble models, we should first focus on developing an excellent individual model, and then improve performance by integrating other learning algorithms.

It can be seen from **Table 3**, the FNN-SVM model, ELM-FNN-SVM model, and MLR-ELM-FNN-SVM model present similar predictive accuracy. Although more learning

algorithms are integrated into ELM-FNN-SVM and MLR-ELM-FNN-SVM models, they did not present significantly better results than the FNN-SVM model, which is caused by the limited predictive ability of MLR and ELM models, deteriorating ensemble performance. In addition, the development of the FNN-SVM model is more efficient for less ensemble computational effort. Therefore, the ensemble model integrated FNN and SVM algorithms are the optimal choice for flash point predictions for its higher accuracy and computational efficiency.

3.3 Evaluation among individual and ensemble models

To make a comprehensive evaluation on the ensemble modeling, predictive performance is compared among 4 individual and 11 ensemble models. It is worth highlighting that all individual and ensemble models are developed on the same training set, and evaluated using the same test set. In **Figure 6**, RMSE values of these models are visualized along with components in each model. In the one-predictor part (i.e., individual predictive models), SVM model presents better accuracy as discussed in previous sections. Therefore, the SVM model is selected as the benchmark model for following evaluations. In ensemble cases, 6 out of 11 models perform better than the SVM benchmark model, including 2 two-predictor, 3 three-predictor, and 1 four-predictor models. It can be seen from **Figure 6**, all these six models integrated the SVM model in ensemble processes. Attributed to the participation of SVM model, all six ensemble models display higher predictive accuracy. However, with one exception, the 8^{th} model (MLR-SVM model) in **Figure 6** also integrates the SVM model, which is expected to be better than the benchmark. However, it shows slightly worse prediction results, which is assumed to be affected by the large predictive errors provided by the MLR model. Therefore, it could be better to get rid of such model with limited predictive capability during ensemble modeling. Nevertheless, the 8^{th} model is still better than these two-predictor models (5^{th}-7^{th} models in **Figure 6**) without coupling the SVM model, indicating the importance of optimal SVM model in this work. Similar conclusions can be draw from the case that only MLR, ELM, and FNN are considered for ensembles, as shown in **Figure S5**.

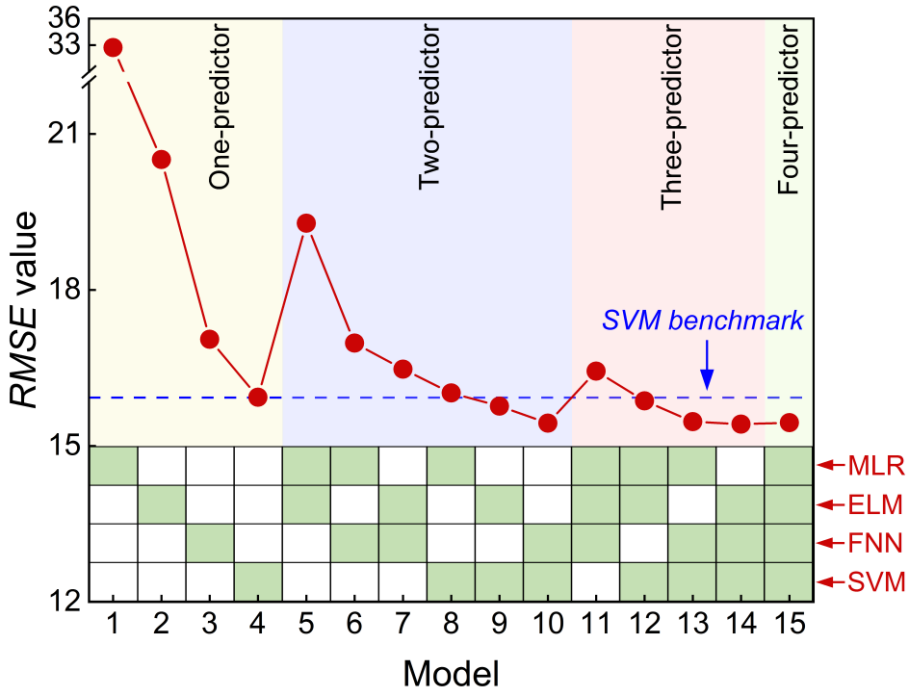

Figure 6. Comparisons among individual and ensemble models (green rectangles indicate activated individual models which participate in the ensembles; for example, the 8th model is MLR-SVM model which is developed by integrating MLR and SVM models).

Moreover, the improvements of model ensembles are quantified by the difference between the individual model and corresponding ensemble model (i.e., $RMSE_{individual} - RMSE_{ensemble}$). From Figure 7, it is observed that almost all of these ensemble cases perform better than the individual cases, indicating the effectiveness of the ensemble learning in improving the model prediction performance. Ensembles on MLR model present the most significant improvements, because of its initial poor predictive accuracy which leaves much room to make progress. In contrast, the SVM individual model already shows high predictive accuracy, and therefore, its ensembles did not present significant progress. Nevertheless, model improvements are achieved by performing the model integration, demonstrating the superiority of ensemble learning assisted property predictive models.

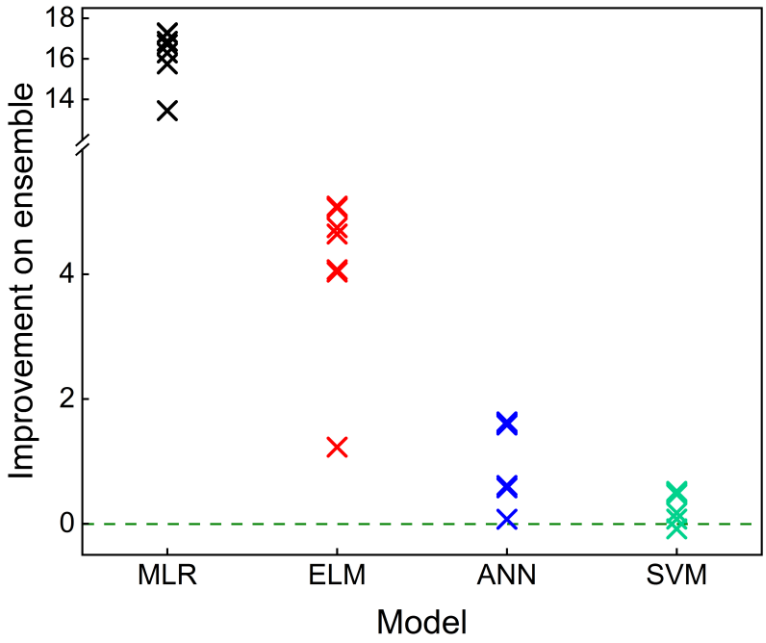

Figure 7. Improvements on predictive models by deploying the ensemble learning.

As the linear ensemble method is employed in this work, it is easy to gain an insight into ensemble learning from the contributions of individual models. Formulas for ensemble models (i.e., 5th-15th models in **Figure 6**) are provided in Formulas (15)-(25), respectively.

$$f(z) = 0.1880 \cdot z_{MLR} + 0.8277 \cdot z_{ELM} - 5.4894 \quad (15)$$

$$f(z) = 0.0317 \cdot z_{MLR} + 0.9775 \cdot z_{FNN} - 3.2685 \quad (16)$$

$$f(z) = 0.1300 \cdot z_{ELM} + 0.8794 \cdot z_{FNN} - 3.3163 \quad (17)$$

$$f(z) = -0.0349 \cdot z_{MLR} + 1.0331 \cdot z_{SVM} + 0.7920 \quad (18)$$

$$f(z) = 0.0794 \cdot z_{ELM} + 0.9238 \cdot z_{SVM} - 0.9680 \quad (19)$$

$$f(z) = 0.6088 \cdot z_{FNN} + 0.3990 \cdot z_{SVM} - 2.6710 \quad (20)$$

$$f(z) = 0.0144 \cdot z_{MLR} + 0.1248 \cdot z_{ELM} + 0.8715 \cdot z_{FNN} - 3.7414 \quad (21)$$

$$f(z) = -0.0375 \cdot z_{MLR} + 0.0828 \cdot z_{ELM} + 0.9546 \cdot z_{SVM} + 0.1824 \quad (22)$$

$$f(z) = -0.0306 \cdot z_{MLR} + 0.6076 \cdot z_{FNN} + 0.4281 \cdot z_{SVM} - 1.7034 \quad (23)$$

$$f(z) = 0.0111 \cdot z_{ELM} + 0.6063 \cdot z_{FNN} + 0.3906 \cdot z_{SVM} - 2.7545 \quad (24)$$

$$f(z) = -0.0311 \cdot z_{MLR} + 0.0142 \cdot z_{ELM} + 0.6044 \cdot z_{FNN} + 0.4178 \cdot z_{SVM} - 1.7949 \quad (25)$$

In most ensemble cases, the FNN and SVM model dominate the development of ensemble models indicated by their larger weights in linear combinations; while MLR and ELM models hold small weights, revealing their insignificant contributions to the ensemble modeling.

Moreover, it is noted that, in some cases, the weights for MLR and ELM models are negative, which means that they are mostly making predictive errors in the same direction as other models holding positive weights. Whereas, with regard to these ensemble models with all positive weights, they are mainly making predictions with error in different directions and achieving the collective-intelligence ensembles by offsetting errors with each other. For better understanding, examples are provided in **Figure S6** to explain the same and different directions in terms of prediction errors.

For these ensemble models integrated both FNN and SVM models (Formulas (20) and (23)-(25)), it is observed that the FNN model holds a larger weight (around 0.6) than the SVM model (around 0.4), which means that the FNN model is much important than the SVM model in these cases. However, as discussed in Sections 3.1 and 3.2, the SVM model performs better than the FNN model, which seems not consistent with the conclusion drawn from the weight values. The inconsistency is caused by their different performance quantified on the training and test sets, respectively. During the model development, both individual and ensemble models are built with the training data, and therefore, the larger weights in ensemble models directly show FNN's better performance with regard to the training set. However, developed models need to be evaluated by the unseen data to quantify their performance in predictions. Thus, when evaluated by the test set, the SVM model performs better than the FNN model, demonstrating its better accuracy in applications.

3.4 Comparison with existing models

The representative individual models (i.e., FNN and SVM models) and ensemble model (i.e., FNN-SVM model) developed in this work are compared to other existing models for flash point predictions, as presented in **Table 4**. Hukkerikar et al.[39] regressed a GC model for flash point prediction using a dataset containing 512 compounds. This GC model employed 93 first-order groups, 62 second-order groups, and 16 third-order groups, presenting satisfactory accuracy with an MAE (mean average error) value of 8.97 and a MAPE (mean absolute percentage error) value of 2.8%. The FNN model proposed by Gharagheizi et al.[40] was developed with 79 functional groups as molecular descriptors (displaying an MAE value of

8.101), while the FNN model developed in this work relies on fewer molecular descriptors (i.e., 69 molecular substructures) showing a smaller MAE value of 7.4508. The SVM model proposed by Bagheri et al.[41] was built with five descriptors derived from molecular structure. However, it exhibits inaccurate results with a large MAE value of 19.31, while the SVM model developed in this work shows a low MAE value of 6.7443. Therefore, our FNN and SVM models are competitive or significantly better than these existing models developed with the same learning algorithms. Furthermore, the FNN-SVM model integrated the FNN and SVM models further improved the model performance, which is also much better than the aforementioned existing models. Individual and ensemble ML models in this work would be more attractive and promising for flash point predictions.

Table 4. Statistical metrics of developed and existing models on the whole dataset.

Model	Method	N	MAE (K)	MAPE	R^2
Hukkerikar et al.[39]	GC	512	8.97	2.8%	0.9671
Gharagheizi et al.[40]	GC-FNN	1378	8.101	-	0.9757
Bagheri et al.[41]	SVM	1651	19.31	5.94%	0.8850
FNN model (this work)	FNN	1732	7.4508	2.0867%	0.9756
SVM model (this work)	SVM	1732	7.1423	2.0716%	0.9743
FNN-SVM model (this work)	FNN, SVM	1732	6.7443	1.9095%	0.9790

Moreover, the lower flammability limit (LFL), which is defined as the lowest concentration of a substance to form a flammable mixture with air, is also used to carry out safety-related properties predictions by the proposed ensemble learning workflow. To build the individual and ensemble models, hyper-parameter optimization, individual model development, and ensemble learning are performed on the LFL dataset containing 1728 data points derived from DIPPR 801 database[35]. More details on ML models for LFL predictions are provided in **Figures S7-S9** and **Table S7**. In addition, ensemble learning on other types of properties is discussed in Pages S9-S10 and S18.

3.5 Model performance regarding molecular structure

The predictive performance of the FNN-SVM ensemble model is further analyzed with regard to different functional-group-based organic compound families, as presented in **Figure 8**. For **Families A** (aliphatic and aromatic hydrocarbons), **B** (alcohols and phenols), **C** (heterocyclic compounds), **G** (esters), **H** (aldehydes), and **J** (organic halogen compounds), R^2 values for both training and test sets are high (over 0.9500), indicating model's reliable and accurate predictions for compounds located within those families. In contrast, the FNN-SVM model shows relatively low R^2 values (below 0.9000) for training and test samples in **Family E** (carboxylic acids). Therefore, it should be noted that predictions could be less reliable when the FNN-SVM model is applied to carboxylic acid compounds.

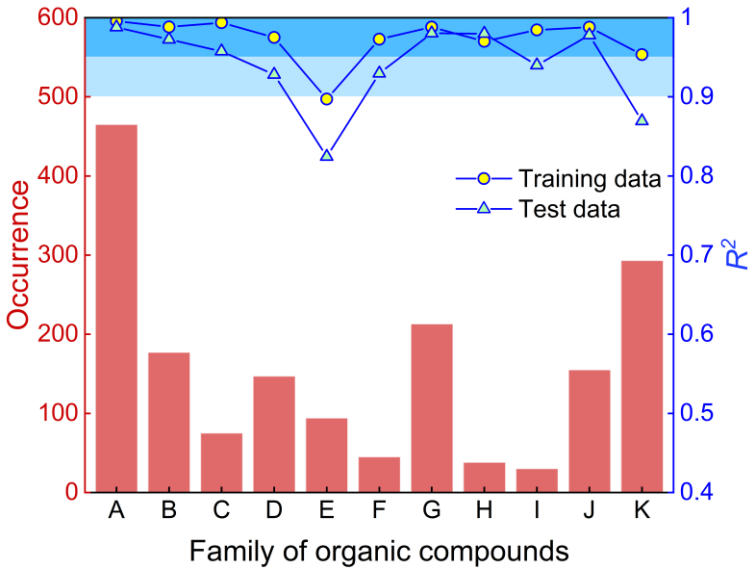

Figure 8. Sample occurrence and model performance regarding the family of organic compound (A: aliphatic and aromatic hydrocarbons; B: alcohols and phenols; C: heterocyclic compounds; D: amines; E: carboxylic acids; F: ketones; G: esters; H: aldehydes; I: ethers; J: organic halogen compounds; K: others).

In some cases, due to the coexist of various functional groups, it could be hard to classify compounds into a definite family. Molecular features are used for model development in this work, and they are highly related to molecular structures. Therefore, a good alternative solution for organic compound classification can be performed on these molecular structure features. To determine the category of organic compounds, t-distributed stochastic neighbor embedding (t-SNE) is employed to reduce the dimensionality of structural features. In this way, the

compressed molecular features are visualized with a two-dimension (2D) embedded space. As shown in **Figure 9**, all investigated compounds are presented in the 2D space and they are colored according to prediction absolute errors. Data points with similar representation variables are concentrated closely, and thereby four groups are formed in the 2D space. For compounds located in **Group 1**, the FNN-SVM model presents a good predictive accuracy (with the RMSE of 8.4335 and R^2 of 0.9857). While predictions for compounds in **Group 3** are relatively unreliable comparing with other groups. Therefore, prediction reliability on new samples can be evaluated by their located groups, which are determined by dimensionality reduction on their structural features.

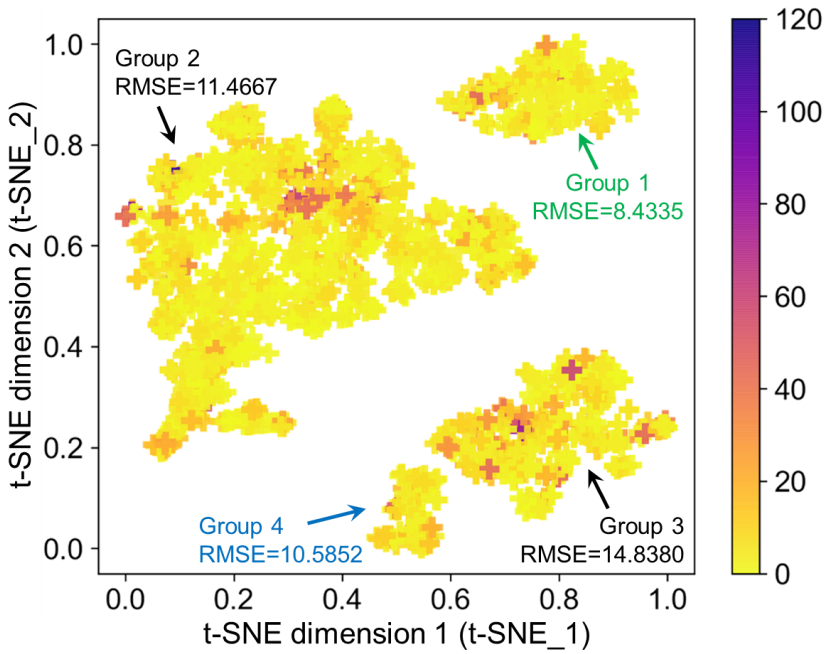

Figure 9. Visualization of chemical compounds in a 2D space, with colors indicating the absolute error of prediction. t-SNE_1 and t-SNE_2 represent two dimensions of the embedded space reorganized into the range [0,1] using the min-max normalization.

4 Conclusions

In this research, ensemble models are deployed on heterogeneous ML algorithms including MLR, ELM, FNN, and SVM. Relying on the case of flash point for chemical substances, individual and ensemble ML models are developed with molecular structure-related descriptors. Among individual models, the SVM model presents the best predictive accuracy. Regarding

ensemble models, they effectively improve the predictive accuracy due to their lower RMSE values than corresponding individual models. Demonstrated by analyses among individual and ensemble models, developing an excellent individual model should take priority, followed by ensemble learning to integrate heterogeneous ML models. Briefly speaking, this research proves that stacking-based ensemble learning is a good choice for ML practitioners to develop models with enhanced predictive accuracy, and guidance for high-efficient development of ensemble-based predictive models is extracted from comprehensive investigations on flash point.

Individual and ensemble ML models in this work can be further used in computer-aided molecular design frameworks to improve the reliability of design procedures in terms of property pre-evaluation. Moreover, the workflow including data processing, feature extraction, predictive modeling, and ensemble learning can be further popularized to other types of properties of interest. Although the proposed FNN and SVM models presented higher accuracy than existing models, the limited predictive capacity of MLR and ELM models hindered the improvement of ensemble models. To overcome this limitation, exploring other promising ML algorithms to discover better models as alternatives could be a practical solution. Moreover, another interesting idea is to perform ensemble learning using a hybrid approach that combines bagging and stacking methods (i.e., integrating homogenous and heterogeneous ML algorithms), as the stacking method focuses on predictive accuracy while the bagging method pays more attention to the reduction of variance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We acknowledge the financial supports provided by the National Natural Science Foundation of China (Grant No. 21878028) and the Chongqing Innovation Support Program for Returned Overseas Chinese Scholars (Grant No. CX2018048).

References

[1] Jhamb, S., Liang, X., Gani, R., Kontogeorgis, G.M., 2019. Systematic model-based methodology for substitution of hazardous chemicals. *ACS Sustainable Chem. Eng*. 7, 7652-7666.

[2] Constantinou, L., Gani, R., 1994. New group contribution method for estimating properties of pure compounds. *AIChE J*. 40, 1697-1710.

[3] Frutiger, J., Marcarie, C., Abildskov, J., Sin, G., 2016. Group-contribution based property estimation and uncertainty analysis for flammability-related properties. *J. Hazard. Mater*. 318, 783-793.

[4] Datta, S., Dev, V.A., Eden, M.R., 2019. Developing non-linear rate constant QSPR using decision trees and multi-gene genetic programming. *Comput. Chem. Eng*. 127, 150-157.

[5] Cao, L., Zhu, P., Zhao, Y., Zhao, J., 2018. Using machine learning and quantum chemistry descriptors to predict the toxicity of ionic liquids. *J. Hazard. Mater*. 352, 17-26.

[6] Liu, Q., Zhang, L., Tang, K., Liu, L., Du, J., Meng, Q., Gani, R., 2021. Machine learning-based atom contribution method for the prediction of surface charge density profiles and solvent design. *AIChE J*. 67(2), e17110.

[7] Su, Y., Wang, Z., Jin, S., Shen, W., Ren, J., Eden, M.R., 2019. An architecture of deep learning in QSPR modeling for the prediction of critical properties using molecular signatures. *AIChE J*. 65, e16678.

[8] Wang, Z., Su, Y., Shen, W., Jin, S., Clark, J.H., Ren, J., Zhang, X., 2019. Predictive deep learning models for environmental properties: the direct calculation of octanol-water partition coefficients from molecular graphs. *Green Chem*. 21, 4555-4565.

[9] Zhong, S., Hu, J., Fan, X., Yu, X., Zhang, H., 2020. A deep neural network combined with molecular fingerprints (DNN-MF) to develop predictive models for hydroxyl radical rate constants of water contaminants. *J. Hazard. Mater*. 383, 121141.

[10] Zhou, T., McBride, K., Linke, S., Song, Z., Sundmacher, K., 2020. Computer-aided solvent selection and design for efficient chemical processes. *Curr. Opin. Chem. Eng*. 27,

35-44.

[11] Zhou, T., Song, Z., Sundmacher, K., 2019. Big data creates new opportunities for materials research: a review on methods and applications of machine learning for materials design. *Engineering* 5, 1017-1026.

[12] Chemmangattuvalappil, N.G., Solvason, C.C., Bommareddy, S., Eden, M.R., 2010. Combined property clustering and GC+ techniques for process and product design. *Comput. Chem. Eng*. 34(5), 582-591.

[13] Zhang, L., Mao, H., Liu, L., Du, J., Gani, R., 2018. A machine learning based computer-aided molecular design/screening methodology for fragrance molecules. *Comput. Chem. Eng*. 115, 295-308.

[14] Zhou, T., Song, Z., Zhang, X., Gani, R., Sundmacher, K., 2019. Optimal solvent design for extractive distillation processes: a multiobjective optimization-based hierarchical framework. *Ind. Eng. Chem. Res*. 58, 5777-5786.

[15] Song, Z., Zhang, C., Qi, Z., Zhou, T., Sundmacher, K., 2018. Computer-aided design of ionic liquids as solvents for extractive desulfurization. *AIChE J*. 64(3), 1013-1025.

[16] Pan, Y., Jiang, J., Wang, R., Cao, H., Cui, Y., 2009. Prediction of the upper flammability limits of organic compounds from molecular structures. *Ind. Eng. Chem. Res*. 48,5064-5069.

[17] Marrero, J., Gani, R., 2001. Group-contribution based estimation of pure component properties. *Fluid Phase Equilib*. 183, 183-208.

[18] Marrero, J., Gani, R., 2002. Group-contribution-based estimation of octanol/water partition coefficient and aqueous solubility. *Ind. Eng. Chem. Res*. 41, 6623-6633.

[19] Jhamb, S., Liang, X., Gani, R., Hukkerikar, A.S., 2018. Estimation of physical properties of amino acids by group-contribution method. *Chem. Eng. Sci*. 175, 148-161.

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

[21] Eslamimanesh, A., Gharagheizi, F., Mohammadi, A.H., Richon, D., 2011. Artificial neural network modeling of solubility of supercritical carbon dioxide in 24 commonly used ionic liquids. *Chem. Eng. Sci*. 66, 3039-3044.

[22] Gharagheizi, F., Eslamimanesh, A., Mohammadi, A.H., Richon, D., 2010. Artificial neural network modeling of solubilities of 21 commonly used industrial solid compounds in supercritical carbon dioxide. *Ind. Eng. Chem. Res*. 50, 221-226.

[23] Pan, Y., Jiang, J., Wang, Z., 2007. Quantitative structure–property relationship studies for predicting flash points of alkanes using group bond contribution method with back-propagation neural network. *J. Hazard. Mater*. 147, 424-430.

[24] Zhou, T., Shi, H., Ding, X., Zhou, Y., 2021. Thermodynamic modeling and rational design of ionic liquids for pre-combustion carbon capture. *Chem. Eng. Sci.* 229, 116076.

[25] Song, Z., Shi, H., Zhang, X., Zhou, T., 2020. Prediction of CO2 solubility in ionic liquids using machine learning methods. *Chem. Eng. Sci*. 223, 115752.

[26] Bhat, A.U., Merchant, S.S., Bhagwat, S.S., 2008. Prediction of melting points of organic compounds using extreme learning machines. *Ind. Eng. Chem. Res*. 47(3), 920-925.

[27] Drucker, H., Burges, C.J., Kaufman, L., Smola, A.J., Vapnik, V., 1997. Support vector regression machines. In: Mozer MC, Jordan MI, Petsche T, eds. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press; 155-161.

[28] Pan, Y., Jiang, J., Wang, R., Cao, H., Cui, Y., 2009. A novel QSPR model for prediction of lower flammability limits of organic compounds based on support vector machine. *J. Hazard. Mater*. 168, 962-969.

[29] Pan, Y., Jiang, J., Wang, R., Cao, H., Cui, Y., 2009. Predicting the auto-ignition temperatures of organic compounds from molecular structure using support vector machine. *J. Hazard. Mater*. 164, 1242-1249.

[30] Svetnik, V., Wang, T., Tong, C., Liaw, A., Sheridan, R.P., Song, Q., 2005. Boosting: an ensemble learning tool for compound classification and QSAR modeling. *J. Chem. Inf. Model*. 45, 786-799.

[31] Varnek, A., Baskin, I., 2012. Machine learning methods for property prediction in chemoinformatics: Quo Vadis?. *J. Chem. Inf. Model*. 52, 1413-1437.

[32] Dev, V.A., Datta, S., Chemmangattuvalappil, N.G., Eden, M.R., 2017. Comparison of tree based ensemble machine learning methods for prediction of rate constant of Diels-Alder reaction. *Comput. Aided Chem. Eng*. 40, 997-1002.

[33] Mayr, A., Klambauer, G., Unterthiner, T., Hochreiter, S., 2016. DeepTox: toxicity prediction using deep learning. *Front. Environ. Sci*. 3, 80.

[34] Zhang, L., Ai, H., Chen, W., Yin, Z., Hu, H., Zhu, J., Zhao, J., Zhao, Q., Liu, H., 2017. CarcinoPred-EL: novel models for predicting the carcinogenicity of chemicals using molecular fingerprints and ensemble learning methods. *Sci. Rep*. 7, 2118.

[35] DIPPR Project 801, Design Institute for Physical Property, AIChE, (2019). https://app.knovel.com/hotlink/toc/id:kpDIPPRPF7/dippr-project-801-full/dippr-project-801-full. Accessed on April, 2, 2019.

[36] Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B.A., Thiessen, P.A., Yu, B., Zaslavsky, L., 2021. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* 49(D1), D1388-D1395.

[37] Wang, Z., Su, Y., Jin, S., Shen, W., Ren, J., Zhang, X., Clark, J.H., 2020. A novel unambiguous strategy of molecular feature extraction in machine learning assisted predictive models for environmental properties. *Green Chem*. 22, 3867-3876.

[38] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint*. 1412.6980.

[39] Hukkerikar, A.S., Sarup, B., Ten Kate, A., Abildskov, J., Sin, G., Gani, R., 2012. Group-contribution+ (GC+) based estimation of properties of pure components: improved property estimation and uncertainty analysis. *Fluid Phase Equilib.* 321, 25-43.

[40] Gharagheizi, F., Alamdari, R.F., Angaji M.T., 2008. A new neural network-group contribution method for estimation of flash point temperature of pure components. *Energy Fuels* 22(3), 1628-1635.

[41] Bagheri, M., Bagheri, M., Heidari, F., Fazeli, A., 2012. Nonlinear molecular based modeling of the flash point for application in inherently safer design. *J. Loss Prev. Process Ind*. 25, 40-51.