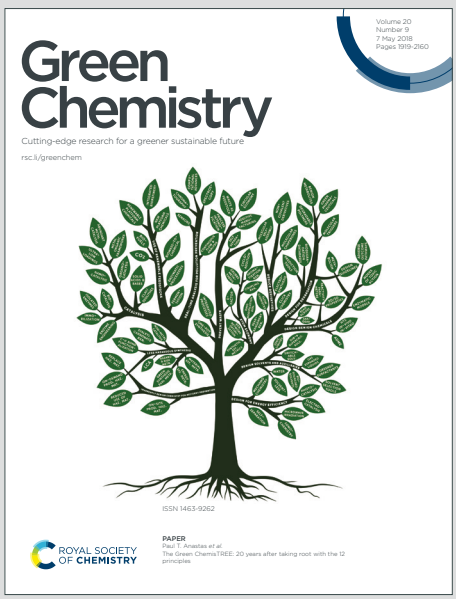
View Article Online
View Journal

Green Chemistry

Cutting-edge research for a greener sustainable future

Accepted Manuscript

This article can be cited before page numbers have been issued, to do this please use: A. Yang, Y. Su, Z. Wang, S. Jin, J. Ren, X. Zhang, W. Shen and J. Clark, *Green Chem.*, 2021, DOI: 10.1039/D1GC00331C.



ROYAL SOCIETY OF CHEMISTRY View Article Online
DOI: 10.1039/D1GC00331C

Green Chemistry

ARTICLE

A multi-task deep learning neural network for predicting flammability-related properties from molecular structures†

Received 00th January 20xx,
Accepted 00th January 20xx

DOI: 10.1039/x0xx00000x

www.rsc.org/

Ao Yang ‡[a,b], Yang Su ‡[c,a], Zihao Wang [a], Saimeng Jin [a], Jingzheng Ren [b], Xiangping Zhang [d], Weifeng Shen *[a] and James H. Clark [e]

It is significant that hazardous properties of chemicals including replacements for banned or restricted products are assessed at an early stage of product and process design. This work proposes a new strategy of modeling quantitate structure-property relationships based on multi-task deep learning for simultaneously predicting four flammability-related properties including lower and upper flammable limits, auto-ignition point temperature and flash point temperature. A multi-task deep neural network (MDNN) has been developed to extract molecular features automatically and correlate multiple properties integrating a Tree-LSTM neural network with multiple feedforward neural networks. Molecular features are encoded in molecular tree graphs, calculated and extracted without manual actions of the user or preliminary molecular descriptor calculation. Two methods, joint training and alternative training, were both employed to train the proposed MDNN, which could capture the relevant information and commonality among multiple target properties. The outlier detection and determination of applicability domain were also introduced into the evaluation of deep learning models. Since the proposed MDNN utilized data more efficiently, the finally obtained model performs better than the multi-task partial least squares model on predicting the flammability-related properties. The proposed framework of multi-task deep learning provides a promising tool to predict multiple properties without calculating descriptors.

1. Introduction

In the last decades, with the growth of chemical production, the potential risks associated with handling hazardous substances have always been of great concern to industry, government and the public. While the production and treatment of (hazardous) chemical substances are now strictly controlled in most regions, legislation affecting the chemicals themselves is only now becoming critical. The advent of Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH) in Europe and similar chemical-focused legislation in other regions including China, Japan and Korea is forcing industry to assess the hazards of all industrial chemicals. Chemicals that are not proven to be sufficiently safe are to be banned or strictly controlled in use. Many widely used chemicals including some of the most important process solvents have been classified in this way, and the list of unacceptable chemicals is increasing rapidly. It is important that the hazardous properties of chemicals involving replacements for banned or restricted products are assessed at an early stage of product and process design.

In Globally Harmonized System (GHS) of classification and labelling of chemicals, four properties including flash point temperature (FPT), auto-ignition temperature (AIT), upper and lower flammability limits (UFL/LFL) are employed to classify chemicals using similar categories.[1] The property, FPT, is often used to evaluate the flammable risk of organic liquid in REACH legislation. AIT is important for the assignment of temperature classes in explosion protection (*i.e.*, ATEX in Europe) of plants and equipment, which can be used to assess situations in which a substance can spontaneously catch fire. UFL and LFL are usually seen as the ease with which a substance can burn or be ignited. The four properties are often used to estimate possibilities of catching fire on substances in many standards and codes.

It is however, very time-consuming to screen safer candidates of hazardous chemicals from many possible compounds through experimental assessments of all the key risk parameters in authorized laboratories. This is considered by many people to be a major disincentive to companies developing new safer chemicals to replace those hazardous compounds judged badly by REACH and related assessments. The GHS rule also states that if experimental values of any properties are unavailable to assess the hazardous level of a chemical, these properties of an individual molecule can be predicted by mathematical models such as quantitative structure-activity/property relationships (QSAR/QSPR). These predictive models could accelerate the process development at least during initial assessments and thus enable early go/no-go decisions in screening of alternatives with lower costs. Among these, the previous QSPR models can estimate properties by using some information of molecular structures, *e.g.*, the occurrences of certain molecular groups[2-4], molecular descriptors and properties[5-7]. Most existing studies of physicochemical properties involving AIT, FPT, UFL and LFL were reviewed by Nieto-Draghi *et al.*[8] and Jiao *et al.*[9]. Herein, Table 1 exemplifies some existing models for predicting the four properties with various features of molecular structures.

Many studies applied group contribution methods and multiple linear regression (MLR) in predicting flammability-related properties, and other models were formulated with various numerical descriptors of molecular

[a] School of Chemistry and Chemical Engineering, Chongqing University, Chongqing 400044, China. Email: shenweifeng@cqu.edu.cn
[b] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China
[c] School of Intelligent Technology and Engineering, Chongqing University of Science and Technology, Chongqing 401331, China
[d] Beijing Key Laboratory of Ionic Liquids Clean Process, CAS Key Laboratory of Green Process and Engineering, Institute of Process Engineering, Chinese Academy of Sciences, Beijing, 100190, China
[e] Green Chemistry Centre of Excellence, University of York, York YO105D, UK
† Electronic Supplementary Information (ESI) available: See DOI: 10.1039/x0xx00000x
‡ These authors contributed equally to this work.

View Article Online
DOI: 10.1039/D1GC00931C

structures. Except for linear models, non-linear regression (NLR) was employed to correlate more flexible models. The stochastic optimization was also used to build prediction models, for the globally optimized correlations *e.g.*, Gharagheizi *et al.*[10, 11], Pan *et al.*[12] and Lazzús[13]. The existing studies based on both MLR and NLR suggested that outlier detection, applicability domain (AD) and uncertainty analysis should be used to evaluate the models[14-17]. Even though this provided a good accuracy for the flammability-related properties, all the studies as listed in Table 1 focused on correlating one property with molecular descriptors in a single task of model regression.

On the other hand, most of previous QSPR studies employed manually-defined methods based on chemistry or graph theory to depict molecular features as numerical descriptors. For example, group contribution methods count occurrences of molecular fragments, but connectivity is frequently ignored among groups. When only one topological index of connectivity is used in a QSPR model, differences among atoms are frequently not recorded. If a new compound has an undefined group or other ambiguous features which cannot be depicted in these manually-defined methods, these models based on only a type of molecular descriptors could not provide a satisfactory estimation. Hence, a variety of descriptors have been developed to enhance the resolution and coverage of diverse molecular structures.[18] As an example, the group contribution plus (GC+) proposed by Hukkerikar[2] combined the multi-level group contribution and atom connectivity indices for wider correlation of more properties. However, it might be time-consuming to select the best descriptors for a task of QSPR modeling.[19, 20]

Table 1. The typical available QSPR-based models for predicting flammability-related properties.

Property	Molecular features	Models	Reference
FPT	The occurrences of molecular groups	MLR	Hukkerikar *et al*.[2]
	The occurrences of molecular groups	MLR	Frutiger *et al*.[3]
	Atom connectivity indices, the occurrences of molecular groups	MLR	Suzuki[21]
	Topological indices	MLR, ANN	Patel *et al*.[5]
	The occurrences of molecular groups	MLR	Alibakhshi[22]
	Molecular descriptors, boiling point	MLR	Katritzky[6]
	The occurrences of molecular groups	GA-MLR	Gharagheizi *et al*.[10]
	The occurrences of molecular groups	ANN	Gharagheizi *et al*.[3]
	The occurrences of molecular groups	SVM	Pan *et al*.[4]
AIT	The occurrences of molecular groups	MLR	Hukkerikar *et al*.[2]
	The occurrences of molecular groups	MLR	Frutiger *et al*.[3]
	The occurrences of molecular groups	ANN	Albahri[23]
	Atom connectivity indices	MLR	Suzuki[24]
LFL/UFL	The number of carbon atoms	NLR	Shimy[25]
	The occurrences of molecular groups	MLR	Frutiger *et al*.[3]
	The occurrences of molecular groups	NLR	Albahri[26]
	The occurrences of molecular groups	ANN	Gharagheizi[27, 28]
	Molecular descriptors	GA-MLR	Gharagheizi[11]
	Topological, charge and geometric descriptors	GA-MLR	Pan *et al*.[12]
	The occurrences of molecular groups	ANN-PSO	Lazzús[13]
	Molecular descriptors	ANFIS	Bagheri *et al*.[29]
	The occurrences of molecular groups	NLR	High and Danner[30]
	The occurrences of molecular groups	MLR	Rowley *et al*.[31]

Abbreviations: genetic algorithm (GA), artificial neural networks (ANN), particle swarm optimization (PSO), adaptive neuro fuzzy inference system (ANFIS), support vector machine (SVM)

A data-driven technique, deep learning, has been recently employed to build QSPR/QSAR models.[32-35] One important reason is that deep learning techniques can extract valuable features automatically and discover potential relationships among various big data. For example, in the Tox21 Data Challenge launched by NIH, EPA and FDA[36], various deep learning neural networks (DNNs) were employed to automatically extract the relevant molecular features from a huge number of descriptors and detect toxicophores. This could help chemists to identify valuable candidates at early stage and with less manual work. As the powerful capabilities of DNNs on extracting features, deep learning techniques can formulate QSPRs/QSARs from visual representations of molecular structures, *e.g.*, images learned by convolutional neural network (CNN)[37], texts learned by recurrent neural network (RNN)[38] and graphs learned by graph neural networks (GAN)[35, 39, 40]. Even though deep learning provides a way to reduce the dependency of QSPRs/QSARs on molecular descriptors[41], the powerful ability could easily make DNNs over-fitted on small data sets. The reason is that DNNs are often formed in the sophisticated architectures involving a very large number of parameters.

Generally, the best way is to train a QSPR/QSAR model based on deep learning using a larger number of samples. Although the prediction models[10, 27, 28] of flammability-related properties performed well, they were trained on more data points including estimated values. In fact, the experimental data sets of four flammability-related properties are available with small sizes in DIPPR801 database according to previous studies[3]. We noticed that multi-task learning (MTL) is a type of transfer learning that can gain relevant knowledge among multiple tasks for modelling on a small data set, even though the relevant knowledge among these tasks may be tenuous and unnoticeable to humans[42]. Caruana[43] studied many technical details in this field and summarized the improvements over single task learning (STL) obtained by MTL: a) amplification of used data; b) attention focusing; c) representation bias and feature selection; and d) regularization against overfitting. Hence, MTL was used to tackle data deficiencies or improve prediction performance for modelling QSARs/QSPRs based on molecular descriptors, *e.g.*, ANN with a multiple output layer[44] and multi-task partial least squares (PLS)[45]. It should be mentioned that the multi-task PLS requires that every molecule in the training set have a complete set of properties. If a particular row of employed data sets has a null value of a feature or a target property, the row corresponding to a compound cannot be utilized in the training of multi-task PLS models.

In this research, an architecture of multi-task deep learning neural networks (MDNN) is proposed to establish the predictive model for flammability-related properties. Molecular structures are transformed into directed acyclic graphs (DAGs) by a program developed in this work, and the DAGs are vectorized by two techniques, word embedding and a tree structured long short-term memory[46] (Tree-LSTM) network. Importantly, unlike previous models, there is no need to extract corresponding features of molecular structures through pre-defined descriptors. The proposed MDNN can extract

molecular features related to a unique property and may capture some possible interactions among multiple properties using two training strategies, *i.e.*, joint training and alternative training. The procedures involving outlier detection and AD analysis are also proposed to assess the predictive model based on multi-task deep learning. All of these are aimed at developing an automatic tool of multi-task QSPR modelling that can simultaneously correlate diverse target properties in one deep learning model without descriptor selections, and fully utilize limited experimental datasets. Another technique of multi-task learning, PLS regression, is taken as a baseline of comparison to highlight better performance and scalable data-handling capacity of the proposed MDNN.

2. Methodology

The entire architecture of MDNN is first introduced which involves two types of modules (see Fig. 1): (1) An encoder based on the Tree-LSTM network[46] that can vectorize the molecular structures depicted in DAG, as well as capture all relevant features and commonality for all tasks; (2) Multiple feedforward neural networks (FNNs) are assigned for extracting task-specific features (learning personality of each task) and outputting each property respectively. Afterwards, the implementation and modelling of MDNN are detailed. In the data preparation, SMILES expressions were converted to DAGs using our program based on Faulon's algorithm[47] in advance. Each vertex of a DAG was labelled in a string involving symbols of atoms and chemical bonds, and the strings were mapped to vectors with the algorithm of word embedding[48]. The procedure had been proposed to transform molecular structures in the previous work[40]. Herein, joint training and alternative training were both employed especially for the multi-task deep learning. The obtained model was finally tested on an external test set to validate the extrapolating ability. Empirical cumulative distribution functions (ECDF) of prediction residuals were employed to detect outliers. The ECDF is a step function that increases by 1/n in every data point. An approach based on principal component analysis (PCA) was proposed to explore applicability domains of the obtained model.

2.1 Network architecture

As mentioned above, the Tree-LSTM network is used as an auto-encoder to extract the holistic features of a molecular structure for all properties. In contrast, several FNNs work similarly to filters which only capture the relevant features for each property. Herein, four FNNs were used to correlate/output the four flammability-related properties and shared one Tree-LSTM network as the encoder of molecular structures.

2.1.1 The extraction block of molecular features. Since Tree-LSTM networks can traverse all vertexes in a DAG and mimic the topological graph of the DAG, the starting point is that molecular structures should be transformed to DAG forms. Although DAGs can be canonized in a certain rule[47], the resulting canonical orientation is still likely to be quite arbitrary among all possible orientations[39]. Hence, in this study, every DAG was generated from each orientation (*i.e.*, traversed from all possible root atoms) and was then vectorized by the Tree-LSTM network respectively. The whole workflow is presented and exemplified by the chemical 2,4,6-Trichlorotoluene (Fig. 2).

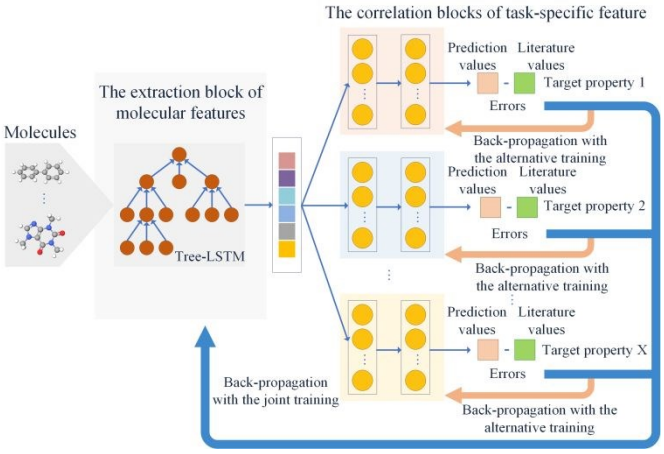


Fig. 1. The schematic diagram of MDNN for modelling QSPRs of multiple target properties.

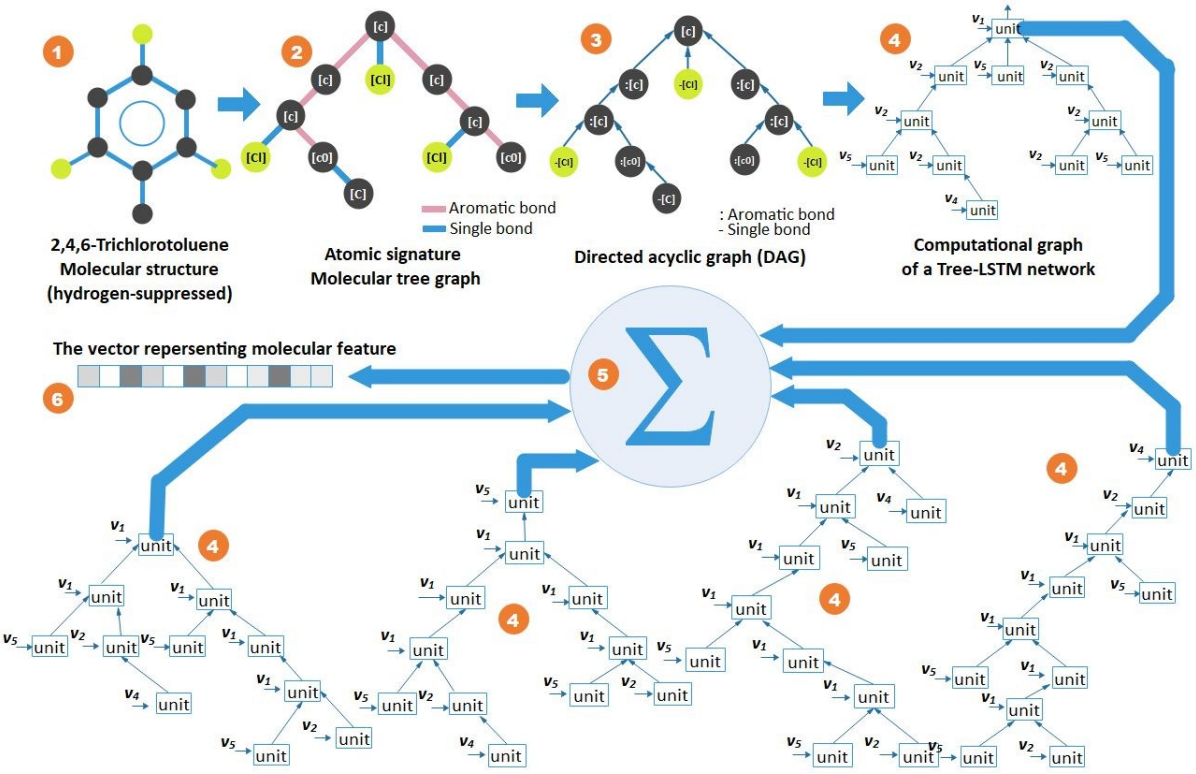


Fig. 2. The molecular structure is transformed to a DAG and then simulated by a Tree-LSTM network.

In the original format of DAGs, each edge does not correspond to a chemical bond of molecules and each vertex is only associated with each atom symbol. For example, a double bond between two atoms cannot be recorded in an original DAG, which is also contrary to the definition of DAGs. To record all bonds of a molecule, a modification was implemented on the original style of DAG, *i.e.*, bond types are attached on adjacent atom symbols (see the third step in Fig. 2) in the children vertexes of a DAG. As the Tree-LSTM network only accepts vectors as inputs of each unit mapping to each vertex of a DAG. As such, the strings (*i.e.*, "[c]", ":[c]", "-[Cl]", "-[C]" and "-[c0]" shown in the third step of Fig. 2) representing the vertexes should be converted to vectors (*i.e.*, v_1~v_5 shown in the fourth step of Fig. 2). A word embedding model, skip-gram[48], was employed to encode such strings as vectors.

Of note is that the Tree-LSTM network can be considered as a dynamic computational graph which has a self-adaptive capability of various molecular structures. For each DAG corresponding to a computation graph of the Tree-LSTM network (see the fourth step in Fig. 2, there are five computation graphs), a recurrent algorithm traverses from the root vertex to leaf vertexes of each DAG and calculate corresponding units of the Tree-LSTM network according to the inputting vectors and neighbourhood outputs. Afterwards, all the vectors representing the DAGs of all orientation are summed into a vector which can represent a molecule (see the fifth and sixth steps in Fig. 2). As space is limited, more details related to the transformation of molecular structures, word embedding and Tree-LSTM are disclosed in the Sections S1 to S3 of Supporting Information.†

2.1.2 The extraction block of task-specific features. Each task-specific block was used to extract relevant features of a specific property and output prediction values. Hence, the number of target properties will determine the number of task-specific blocks, *i.e.*, the number of tasks. Theoretically, each task-specific block can be designed as a different FNN with the independent structure and parameters, and it is also workable to train each block with different optimizers respectively. In the task-specific block, an activation function, rectified linear (ReLU)[49], was applied to perform non-linear transformations and generate activation values, since ReLU has lower computation consumptions and lower risks of gradient vanishing. Although the scalable architecture can provide flexible configurations for multiple tasks of modelling QSPRs, it becomes more challenging to optimize more hyper-parameters. Herein, four FNNs corresponding to the four flammability-related properties (i.e., FPT, AIT, UFL, LFL) were configured with same structural parameters for lower complexity of optimization.

2.2 MDNN implementation and training

As the proposed MDNN includes a dynamic neural network (*i.e.* Tree-LSTM network), it was implemented on an open-source software platform supporting dynamic networks, PyTorch[50]. All tasks of training, validation and testing were finished on the hardware platform with NVidia GTX1060 and Intel i5 8400. A parser based on RDKit[51] was developed and Faulon's algorithm[40] was implemented in Python, to translate SMILES expressions into DAGs. Meanwhile, a simple implementation of word embedding algorithm[52] was utilized to train the embedding vectors in TensorFlow[53]. After all programs were prepared, a multi-task prediction model of the four flammability-related properties was obtained using the procedure illustrated in Fig. 3.

In the workflow of Fig. 3, a regression algorithm named Adam[54] was employed as optimizers to train the proposed MDNN. Eight optimizers were configured with different hyper-parameters for each task in alternate training. Meanwhile, another optimizer was employed to carry out the joint training on the entire MDNN. Additionally, early stopping was used to avoid overfitting, *i.e.*, once there was no improvement on loss values of validation sets for a specified time (*e.g.*, twenty epochs), the training process would be terminated.

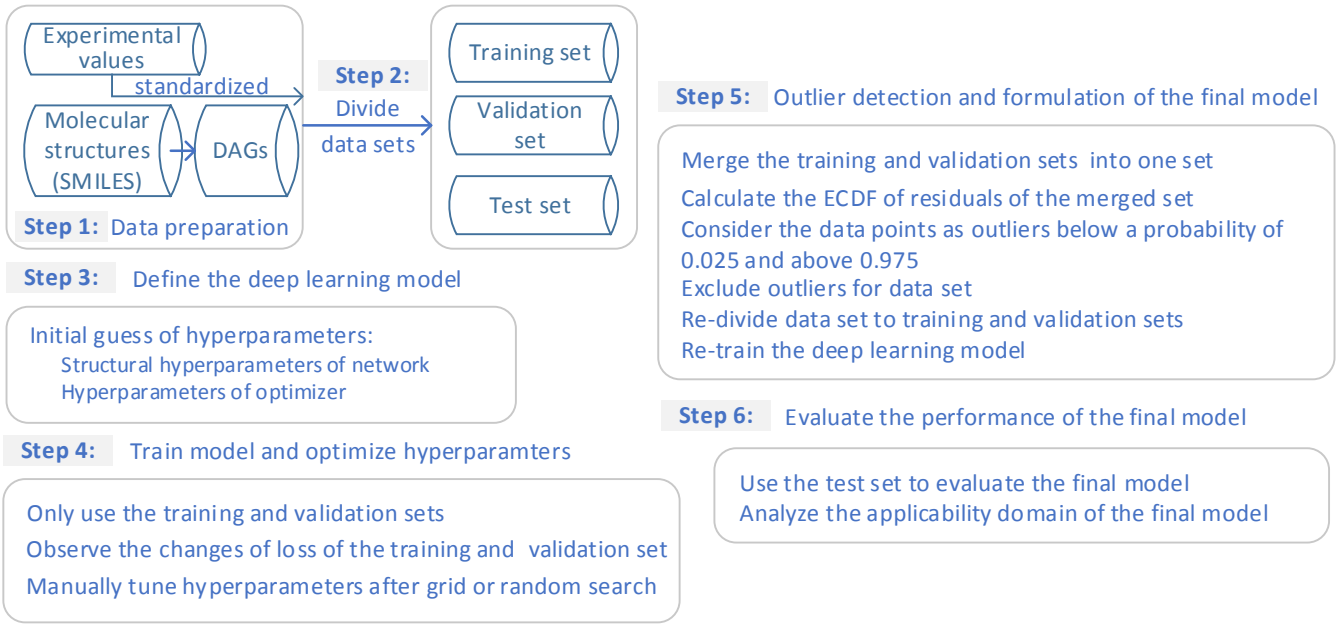


Fig. 3. The overview of the methodology for formulating a multi-task QSPR model based on deep learning.

2.2.1 Data analysis and preparation. The experimental data involving FPT, AIT, LFL and UFL were used to train the MDNN, which were extracted from an authoritative database named DIPPR801[55]. The database, DIPPR801, provides the uncertainty and acceptance for each experimental data point in which all data had been checked and reviewed by database maintainers. Hence, it was employed in many QSPR studies as a reliable data source. A list of molecular structures was gathered from PubChem[56] representing isomeric SMILES, including all the available compounds in DIPPR801 and other compounds. Some compounds involving inorganic gas, salts, metal-organics and metallic elements were excluded from the employed data set, as molecular structures of these unemployed compounds are significantly different from most conventional organics and their flammability property data are often unavailable. The employed datasets only including accepted and experimental values were stored in (Comma-Separated Values) CSV format. The lists of employed compounds are provided in the Section S4 of Supporting Information. †

As for different units and numerical levels among the four flammability-related properties, it is necessary to standardize the raw data sets for easier training models. The reason is that the different numerical metrics could result in big gaps of gradient among each task. Note that the raw experimental datasets should be standardized in a linear transformation to guarantee that distribution shapes are not changed. The Z-score transformation as shown in

Eq. (1) was employed that can always produce a distribution with a mean of 0 and a standard deviation of 1.

$$\hat{x}_i^{\exp} = \frac{x_i^{\exp} - \overline{x}^{\exp}}{\sigma} \quad (1)$$

where $\hat{x}_i^{\exp}$ represents the transformed property values and $x_i^{\exp}$ refers to the original values, $\overline{x}^{\exp}$ and σ are the average value and standard deviation of a data set, respectively. It should be noticed that the $\overline{x}^{\exp}$ and σ must be calculated on training sets, then two parameters are used to transform the validation and test sets.

All available data were employed in the initial correlation, in which outliers were not excluded. The final correlation based on the proposed MDNN was implemented without outliers. Tables 2 and 3 list the used data points for the flammability-related properties in the initial and final correlation, respectively. Table 4 shows information related to the ranges of raw data collected from DIPPR801 database. For training and optimizing the multi-task model, the data set was split into three sets including a training set, a validation set and a test set. The transformation should be only implemented on the training set of each property. The term "validation set" refers to the dataset utilized to determine the hyper-parameters and observe trends of loss for early stopping in training models. The test set was employed to test model performance finally after a deep learning model was well trained with determined hyper-parameters. All compounds both in the validation and test sets were sampled randomly.

Before proceeding with the training, all molecular structures were converted into DAGs attached embedding vectors as mentioned above. Since the generation of embedding vectors does not need property data, many more compounds can be employed by the word embedding algorithm. In this work, 23709 compounds depicted in SMILES expressions were employed to train the embedding vectors for each vertex of DAGs. At this stage, 170 symbols were extracted from all vertexes of the DAGs and each one was represented by a 48-dimensional vector. The symbols are listed in the Section S2 of Supporting Information. †

Table 2. Data points of the flammability-related properties used in the initial correlation.

Property	Whole dataset	Training set	Validation set	Test set
FPT	1176	822	177	177
AIT	501	349	76	76
LFL	449	315	67	67
UFL	350	243	53	54

Table 3. Data points of the flammability-related properties used in the final correlation.

Property	Whole dataset	Training set	Validation set	Test set
FPT	1176	822	177	177
AIT	480	334	70	76
LFL	443	309	67	67
UFL	329	226	49	54

Table 4. Ranges of used datasets on the flammability-related properties.

Property	minimum	maximum	average	standard deviation
FPT	87.1	570	330	63.4
AIT	363	1283	651	120
LFL	0.0454	16.9	2.24	2.41
UFL	2.40	100	12.7	9.31

2.2.2 Training and evaluating a deep learning based QSPR model. Unlike single-task DNNs, there are two strategies to train multi-task DNNs as usual, *i.e.*, alternate training and joint training.[57] With joint training for property prediction, a distribution of vectors representing molecules in the chemical latent space is simultaneously organized by the four properties and commonalities among the four tasks which could be learned. There is a requirement on the training data for joint training, *i.e.*, the experimental values of four properties must be available simultaneously in one data row. Therefore, all parameters of the MDNN can be trained in each epoch of the joint training. After the joint training had been conducted, alternate trainings were employed to train each FNN as well as the Tree-LSTM network in one epoch. It is not necessary to fill all training data matrices in the alternate training which could transfer some information from rich data sets to sparse ones. In the alternate training, only the corresponding FNN is updated for the current task during each iteration, while all parameters of other task-specific FNNs are frozen. Two types of loss function were employed in the two types of training methods. For the joint training, all parameters of the whole network were updated to minimize a combined loss function Eq. (2).[58]

$$\mathcal{L}(\boldsymbol{W}, \sigma_1, \sigma_2, ..., \sigma_i) = \sum_i \left(\frac{1}{2\sigma_i^2} \mathcal{L}_i(\boldsymbol{W}) + \log \sigma_i \right) \quad (2)$$

$$\mathcal{L}(\boldsymbol{W}, \sigma_i) = \frac{1}{2\sigma_i^2} \mathcal{L}_i(\boldsymbol{W}) + \log \sigma_i \quad (3)$$

where $\boldsymbol{W}$ is used to represent model parameters and σ_i is the observation noise parameter (*i.e.*, standard deviation) of each task. This loss function weighs the losses of all tasks using the homoscedastic uncertainty of each task, which allows each task corresponding to each property to be learned simultaneously. $L_i(\boldsymbol{W})$ representing the loss function for each task was calculated by mean square error (*MSE*) between the estimated values and the experimental values. It should be noted that the loss function $L_i(\boldsymbol{W})$ was employed for each FNN and the weighted loss function $L_i(\boldsymbol{W}, \sigma_i)$ was used to train the Tree-LSTM encoder in the alternate training. The details related to model training are presented in the Section S5 of Supporting Information.†

2.2.3 Determination of hyper-parameters. It is significant to determine the optimal configuration of hyper-parameters for a deep learning model. In general, two parts of hyper-parameters should be determined: the structural parameters of deep neural network (*e.g.*, number of hidden layers, number of neurons, types and parameters of activation functions) and parameters of training optimizers. Unfortunately, there is no such a universal configuration that can work well for all models and data sets. It is necessary to optimize the hyper-parameters for different models separately.

In most cases, two approaches, grid search and random search, can provide an acceptable configuration of hyper-parameters for a certain data set. Since there exist 19 hyper-parameters in the proposed MDNN, it is extremely time-consuming to assess all possible combinations of hyper-parameters. In the training process of the proposed MDNN, an initial guess of structural parameters is chosen to fix the network architecture referred to the successful practices in previous studies[35, 58, 59]. To reduce the complexity of hyperparameter optimization, the numbers of neurons in the hidden layers were scanned while the layer number of each task-specific FNN was fixed at four (three hidden layers and one output layer), since there were no significant improvements for more layers. Afterwards, a grid search was applied to find the optimal ranges of other hyper-parameters initially. The hyper-parameters were also fine-tuned into the optimal range manually. Herein, one or two hyper-parameters were tuned at a time manually, it should be observed whether the MDNN performance was improved on the validation sets after manual tuning. The finally adopted configuration of hyper-parameters are listed in Tables 5 and 6. After all the hyper-parameters were determined and validated, the model was tested on an external test set eventually.

Table 5. The finally adopted structural hyper-parameters of the MDNN.

Hyper-parameters	Values
The dimension of embedding vectors	48
The memory dimension of the Tree-LSTM	32
The output dimension of the Tree-LSTM	32
The hidden layers of each task-specific FNN	3
The neuron number of each hidden layers in the FNNs	32

2.2.4 Statistical evaluation. The following statistical indicators were employed as the performance metrics to evaluate the learned MDNN model.

Mean absolute error (*MAE*) is the measure of deviation between the predicted values and the experimental values, and it is obtained *via* Eq. (4).

$$MAE = \frac{\sum_{i=0}^{N} | x_i^{\exp} - x_i^{\mathrm{pre}} |}{N} \quad (4)$$

Mean percentage error (*MPE*) provides an average of percentage error by which the predicted values differ from the experimental values, and it is expressed as Eq. (5).

View Article Online DOI: 10.1039/D1GC00331C

Table 6. The finally adopted hyper-parameters of the training optimizers.

Hyper-parameters	Values	Tasks
The learning rate η_t	0.02000	Jointly, training the whole MDNN according to all properties
The learning rate η_1	0.00120	Alternatively, training the Tree-LSTM network according to FPT
The learning rate η_2	0.00120	Alternatively, training the Tree-LSTM network according to AIT
The learning rate η_3	0.0006	Alternatively, training the Tree-LSTM network according to LFL
The learning rate η_4	0.00023	Alternatively, training the Tree-LSTM network according to UFL
The learning rate η_1	0.00120	Alternatively, training the task-specific FNN according to FPT
The learning rate η_2	0.00180	Alternatively, training the task-specific FNN according to AIT
The learning rate η_3	0.00009	Alternatively, training the task-specific FNN according to LFL
The learning rate η_4	0.00020	Alternatively, training the task-specific FNN according to UFL
The batch size b_t	32	Jointly, training the whole MDNN according to all properties
The batch size b_1	32	Alternatively, for the Tree-LSTM and FNN corresponding to FPT
The batch size b_2	32	Alternatively, for the Tree-LSTM and FNN corresponding to AIT
The batch size b_3	32	Alternatively, for the Tree-LSTM and FNN corresponding to LFL
The batch size b_4	32	Alternatively, for the Tree-LSTM and FNN corresponding to UFL

$$MPE = \frac{1}{N}\sum_{i=0}^{N}\frac{\left|x_i^{\exp} - x_i^{\mathrm{pre}}\right|}{x_i^{\exp}} \times 100\% \quad (5)$$

Since the correlation analysis reported in the published literatures[21, 22, 24, 32, 33, 60, 61] usually chose the correlation coefficients of r or R^2 as the performance indicator, both of them were employed to assess the proposed MDNN in this research.

The Pearson correlation coefficient,

$$r = \frac{\sum_{i=1}^{n}(x_i^{\exp} - \overline{x^{\exp}})(x_i^{\mathrm{pre}} - \overline{x^{\mathrm{pre}}})}{\sqrt{\sum_{i=1}^{n}(x_i^{\exp} - \overline{x^{\exp}})^2}\sqrt{\sum_{i=1}^{n}(x_i^{\mathrm{pre}} - \overline{x^{\mathrm{pre}}})^2}} \quad (6)$$

The coefficient of determination,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i^{\exp} - x_i^{\mathrm{pre}})^2}{\sum_{i=1}^{n}(x_i^{\exp} - \overline{x^{\exp}})^2} \quad (7)$$

2.3 The outlier detection and AD determination

The outlier detection was also studied for deep learning based QSPRs in this work. The outliers of four tasks were detected with the empirical cumulative distribution function (ECDF) of the residuals between experimental and predicted values. The ECDF is a step function that increases by 1/n in every data point. Let (X_1, …, X_n) be independent, identically distributed real random variables with the common cumulative distribution function F(t), then the ECDF can be defined as shown in Eq. (8) for a realization (x_1, ..., x_n).

$$\mu_n(t) = \frac{\text{number of element in the sample} \le t}{n} = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \le t} \quad (8)$$

For a fixed t, the indicator $I_{X_i \le t}$ is a Bernoulli random variable with parameter $p=F(t)$. The value of the ECDF is the number of samples whose sample value is less than or equal to t divided by the total number of samples n. This methodology was suggested in the residual analysis of group contribution models by Frutiger *et al*.[14], in which the distribution of residuals could not be assumed as a normal distribution in advance. Frutiger *et al*.[14] applied the approach on a unsegmented data set and repeatedly regressed a group contribution model after outliers were excluded. Finally, the model performance was improved successfully thanks to the reduction of residual dispersion. However, raw data sets are often divided into three subsets for training, validation, and test respectively in the modelling of QSPRs based on deep learning. Although outliers can be detected on all the three data sets, only the outliers in the training and validation sets can be excluded. The test set is often utilized to evaluate the generalization ability of a trained model of deep learning, which is independent to the training and validation sets and should not be evaluated in the model training. Moreover, there is a possibility that some compounds included in test sets could drift out of the latent chemical space determined by training and validation sets. To depict the latent chemical space learned by MDNN and check the changes of chemical space after outliers are excluded, we also investigated ADs of the final model from two aspects: structural domain and property domain.[16]

Molecular features are depicted in a high-dimensional vector output by the Tree-LSTM network. As the high-dimensional vector cannot be visualized and analysed easily, it is suitable to apply a dimension-reduction algorithm, principal components analysis (PCA), on the determination of ADs involving training and validation sets[62]. We decided to assess structural space and property space together, *i.e.*, all the high-dimensional vectors and target properties in a dataset were combined into a matrix. PCA can reduce dimensionality and enable only a few principal components (PCs) to retain the most of variance of all data. As such, ADs could be analysed and visualized in a lower dimensional space. Another factor is that the scope of an AD should be also determined, a strategy based on convex hull was applied to explore the AD boundary. For this reason, it is easy to discover whether some compounds in test sets exist outside the AD. Meanwhile, the outliers identified by ECDF can also be marked in a visualized AD, to investigate relationships between the outliers and ADs. The calculation method of convex hull is disclosed in Section S7 of the Supporting Information. †

3. Results and discussion

After the time-consuming tuning of hyperparameters was finished in the MDNN training, the obtained model was evaluated on the data sets. For each property, the model performance was measured in four statistical indicators involving *MAE*, *MPE*, *r* and R^2. The identified outliers are provided in Table S10 of the Supporting information. †Another multi-task learning algorithm, partial least square (PLS) [45], were employed to compare with the proposed MDNN. Comparisons with the existing predictive models of flammability-related properties are provided in Section S6 of the Supporting information. †

Two results are presented for two models obtained before and after outliers were excluded, respectively. Fig. 4 shows prediction deviations obtained by the model (Ⅰ) trained with all data points. After the outliers were identified and excluded, the proposed MDNN was re-trained without the outliers of training and validation sets. Outliers were identified for the retrained model (II) again, Fig. 5 provides predictions of the model (Ⅱ). In Figs. 4 and 5, diagonal lines represent the equivalence between experimental values and

predicted values, while circles, rectangles, and plus signs indicate the predicted values of compounds in training, validation, and test sets, respectively. Tables 7 and 8 present the values of performance metrics for the models (Ⅰ) and (Ⅱ) respectively. It suggested that the proposed MDNN model can accurately predict the four flammability-related properties with small deviations for most compounds.

The relationships between compound families as well as prediction errors have also been investigated according to the model (I). Distributions of compounds in various families were presented with MPE in Figs. S15~S18 of the Supporting Information. For each property, the MPE of each family was calculated on a union set of training, validation and test sets. Since a method of random sampling was used to divide data sets, some families with much less compounds would be not selected into the validation and test sets. As shown in Fig.S15~S18, chemicals in training sets are more diverse than validation and test sets. For example, several compound families (*e.g.*, 1-alkenes, alkynes, formates, etc.) were not sampled into the validation and test sets for FPT, *i.e.*, the diversity of training set is larger than that of validation and test set. Among the four properties, compounds correlated with FPT distribute in 79 families, the least number of families employed in the correlation of UFL is 67. Molecules used to train MDNN on AIT distribute in 73 families, and the compounds used in the correlation of LFL distribute in 72 families.

Although the MDNN model predicted FPT accurately on most compounds, the MDNN model would produce a large prediction error on a small molecule, for example, methane which only has one carbon atom. The model provided a good value of *MPE* within 10% on halogenated hydrocarbons only including a single atom of halogen. However, a large prediction error output by the MDNN model would show on the hydrocarbons halogenated with two different halogen atoms. Furthermore, if C=C double bonds existed in a molecule with halogen atoms together, the model would give a larger prediction error of FPT. The reason could be due to substitution positions of halogen atoms in halogenated olefins, which cause significant changes of chemical properties of these molecules. On the other hand, only a small number of halogenated olefins were available in the DIPPR801 database, MDNN cannot learn enough information of this compound family. For other compounds containing other heteroatoms, such as O,N,S,P,Si and other atoms, the model can also predict FPT accurately on most of these compounds. It can be found that the outliers are mainly concentrated in a few of compounds with multiple functional groups, such as dinitrobenzene, ethylene glycol, dimethyl chlorosilane. According to the AD analysis, 8 compounds were predicted outside the application domain, but only one compound, hexachlorobenzene was identified as an outlier because of its large relative error (19.8%). The reason could be only 1 to 3 chlorine atoms exist in aromatic hydrocarbons of the training set.

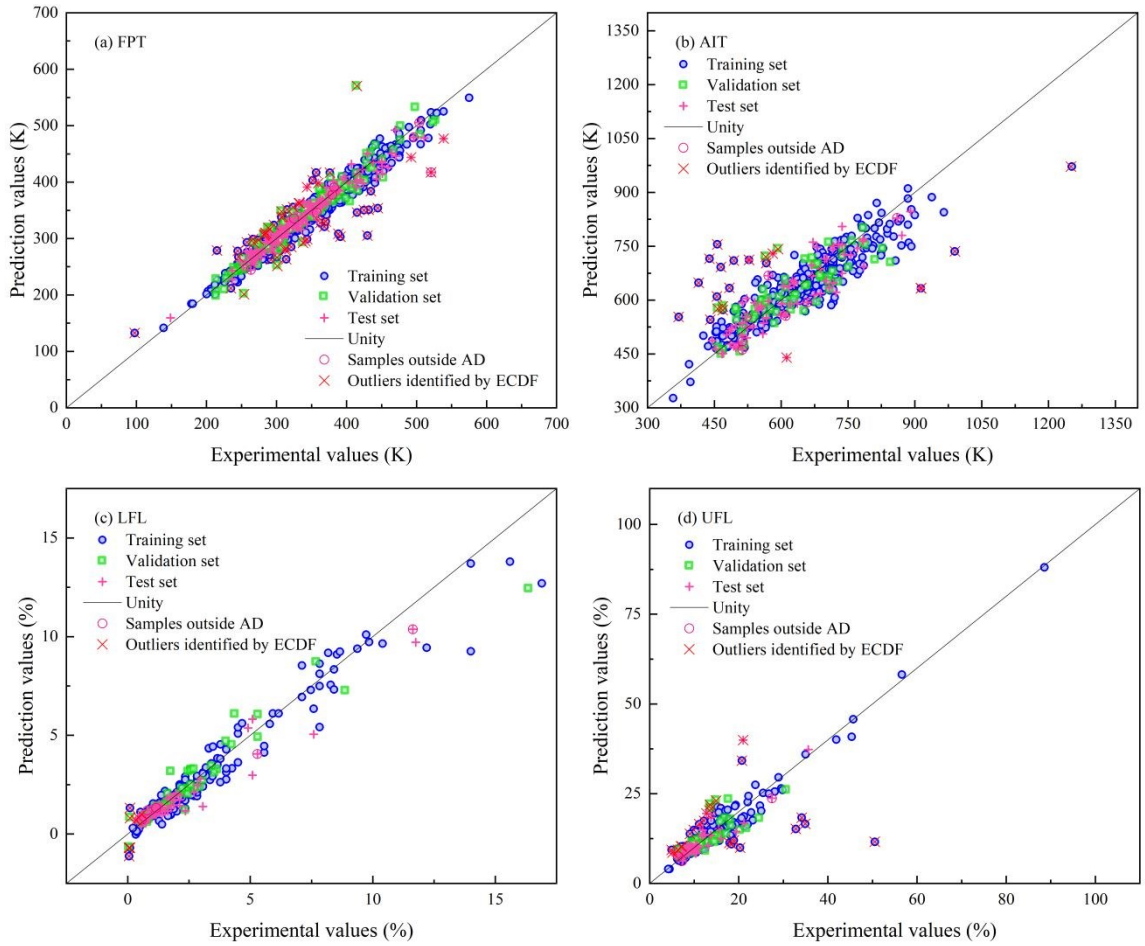


Fig. 4. The experimental versus predicted values of four flammability-related properties for the training, validation and test sets obtained by the model (I): (a) FPT; (b) AIT; (c) LFL; (d) UFL.

The model can predict AIT as accurately as existing models. A large prediction error was observed mainly on peroxides, since only two compounds can be learned by MDNN in the peroxide family. In the other words, not enough samples involving the molecular feature "-O-O-" could be learned by MDNN. The MDNN model mainly predict LFL inaccurately on three compound families including nitriles, monoaromatics and aromatic esters. Five compounds were identified as the samples outside the AD of training set of LFL, i.e., isobutyl acrylate, benzonitrile, tetrafluoroethylene, hydrazine, and trans-decahydronaphthalene. For example, the model can estimate an acceptable value of LFL on tetrafluoroethylene, although MDNN did not learn the molecule including C=C bonds and four fluorine atoms. Benzoyl chloride and difluoromethane were determined as the samples outside AD of training

View Article Online
DOI: 10.1039/D1GC00532C

set of UFL, their relative errors were smaller than 20%. The highest prediction error of MDNN appeared on the families including mercaptans and polyfunctional C, H, O, N for UFL. Only two compounds of mercaptans can be used to train MDNN for predicting UFL. Another two compounds involving N-methyl-2-pyrrolidone and morpholine in the family of polyfunctional C, H, O, N were predicted with the high relative errors for UFL. In short, the MDNN model could show more prediction errors on polyfunctional molecules.

Among four flammability-related properties, the learned models (Ⅰ) and (Ⅱ) both provide the highest accuracy on FPT. For AIT, the values of *MPE* are also acceptable for the test set, however, the data points show more dispersion with smaller R^2 and larger *MPE* than other properties. For LFL and UFL, these two models also provide an acceptable accuracy for most compounds while a few compounds deviate from the diagonal lines as shown in Fig. 4(c, d) and Fig. 5(c, d). It is worth noting that the magnitudes of LFL and UFL are smaller than those of FPT and AIT. Since the smaller absolute values tend to result in larger deviation in *MPE*, outliers can result in the higher MPE and lower R^2 especially for LFL. As the distribution of raw data points of AIT are more disperse than those of other properties, these two models perform unexpectedly on AIT. However, the models can still predict properties precisely on the test sets. It suggested that the correlations between properties and molecular structures were learned by the proposed MDNN without calculating descriptors. Four empirical cumulative distribution functions (ECDFs) of prediction residuals are described in Fig. 6 for the model (Ⅰ). The two horizontal lines at the bottom and top of each subplot of Fig. 6 represent the probabilities, 0.025 and 0.975, respectively. Data points that are not reasonably likely expected to occur according to the empirical CDF can be considered as outliers, i.e. data points can be considered as outliers below the 0.025 or above the 0.975 probability levels. The list of outliers is presented as Table S10 of the Supporting Information. † As shown in Table 8, the model (Ⅱ) performs better on the training and validation sets for FPT, AIT, and LFL according to *MPE*. The higher values of *MPE* is obtained by the model (Ⅱ) on UFL, despite higher values of *r* and R^2. Comparing Table 7 and Table 8, we can see that the exclusion of outliers does not always provide better results for the deep learning model. Except for UFL, the model (II) provides worse performance on the test sets of other properties.

In this research, our interest is particularly focused on the changes of ADs caused by the exclusion of outliers. The matrix of an AD was reduced into a three-dimensional (3D) space for visualization, consisting of the molecular feature vectors and target properties of molecules. The three principal components (PCs) can explain more than 85% of variances in the raw space of training sets (33 dimensions). After the convex hull of AD was determined on training and validation sets, the compounds which may appear outside the AD were identified from a test set. For the model (Ⅰ), Fig. 7 presents the scatters of three PCs for training sets, validation sets, and test sets of the four properties, respectively. Meanwhile, the outliers are also marked in Fig.7, which was identified by ECDFs. The visualized ADs of models (Ⅰ) and (Ⅱ) are shown as 3D convex hulls in Figs. S11-S14 of Supporting Information. †

A convex hull defines an interpolation region formulated by experimental data and molecular feature vectors for a property. The boundary of a convex hull describes the smallest convex area covering a training set and the corresponding validation set. When a compound appears outside the convex hull, the compound will be extrapolated by the model built on the training and validation sets. Notably, the model has abilities to predict flammability properties on compounds outside ADs. For example, when the proposed model predicted FPT on squalene, the model still provided an accurate result (only 3.33% relative errors), but the model did not learn the complicated molecular structure which included so many carbon atoms.

When the convex hulls and outliers are observed together (see Figs. S6-S9 of Supporting Information †), more outliers appear near the boundaries of convex hulls or in the low-density region of scatters. Once the outliers were excluded according to residual ECDFs (see Fig. S10 in the Supporting Information†), the convex hulls would cover smaller ranges. The ADs of LFL and FPT became significantly narrower, while the ADs of AIT and UFL changed slightly. As discussed earlier, the exclusion of outliers did not result in better performance of the retrained model on test sets. In other words, the generalization ability of retrained models could be declined since less samples remains in the downsized training sets. It can be also observed that the convex hulls representing ADs also include some considerable empty space. If more outliers are blindly excluded, the ADs will be narrower and the empty space will be reduced. Although not each point in the convex hulls can correspond to a potentially feasible molecular structure, the outlier exclusion still may reduce the interpolation space of learned models and make predictive models meaningless. As the empty space a learned model of interpolation covers, declines, so does the need to include property-space in the assessment.

Table 7. The performance statistics of the initially learned MDNN model (Ⅰ)

		FPT (K)	AIT (K)	LFL (%)	UFL (%)
MAE	Training set	10.17/8.448[a]	38.14/32.41[a]	0.2918/0.2867[a]	2.032/1.472[a]
	Validation set	12.27/10.51[a]	49.59/41.16[a]	0.3564/0.3449[a]	2.055/1.686[a]
	Test set	10.96/10.45[b]	45.34/44.76[b]	0.3663/0.3445[b]	1.815/1.806[b]
MPE	Training set	2.990/2.449[a]	6.069/4.925[a]	22.76/14.51[a]	14.50/11.75[a]
(%)	Validation set	3.575/3.046[a]	8.143/6.409[a]	48.28/12.82[a]	14.77/11.68[a]
	Test set	3.101/3.010[b]	7.218/7.071[b]	19.52/19.14[b]	15.39/15.40[b]
r	Training set	0.9712/0.9856[a]	0.8757/0.9302[a]	0.9728/0.9736[a]	0.8880/0.9699[a]
	Validation set	0.9561/0.9779[a]	0.7911/0.8847[a]	0.9608/0.9617[a]	0.7947/0.8920[a]
	Test set	0.9660/0.9698[b]	0.8305/0.8237[b]	0.9702/0.9558[b]	0.8582/0.8521[b]
R^2	Training set	0.9394/0.9680[a]	0.7632/0.8550[a]	0.9449/0.9464[a]	0.7883/0.9404[a]
	Validation set	0.9060/0.9538[a]	0.6256/0.7623[a]	0.9164/0.9171[a]	0.6114/0.7602[a]
	Test set	0.9303/0.9387[b]	0.6778/0.6780[b]	0.9138/0.8828[b]	0.6134/0.5294[b]

NOTE: [a] This value was obtained on the data set without the outliers identified by ECDF; [b] This value was obtained on the data points only involved in the AD determined by training and validation sets.

Table 8. The performance statistics of the finally re-trained MDNN model(Ⅱ)

		FPT (K)	AIT (K)	LFL (%)	UFL (%)
MAE	Training set	10.15/8.995[a]	34.02/30.21[a]	0.2398/0.3207[a]	2.366/1.759[a]
	Validation set	11.49/9.695[a]	40.65/32.74[a]	0.2772/0.3066[a]	3.026/2.430[a]
	Test set	13.56/12.59[b]	56.12/54.34[b]	0.3881/0.2617[b]	2.439/2.505[b]
MPE	Training set	3.120/2.729[a]	5.367/4.519[a]	16.27/17.77[a]	18.77/13.46[a]
(%)	Validation set	3.401/2.937[a]	6.443/5.490[a]	19.87/12.10[a]	18.99/17.32[a]
	Test set	4.001/3.762[b]	9.234/8.889[b]	19.26/16.77[b]	24.15/24.45[b]
r	Training set	0.9808/0.9871[a]	0.9421/0.9204[a]	0.9754/0.9880[a]	0.9623/0.9633[a]
	Validation set	0.9643/0.9836[a]	0.9135/0.9389[a]	0.9809/0.9812[a]	0.7832/0.8608[a]

	Test set	0.9554/0.9595[b]	0.8020/0.8015[b]	0.8876/0.9736[b]	0.8752/0.8490[b]
R^2	Training set	0.9541/0.9775[a]	0.8845/0.9067[a]	0.9877/0.9586[a]	0.9238/0.9693[a]
	Validation set	0.9214/0.9616[a]	0.8173/0.8778[a]	0.9619/0.9625[a]	0.5703/0.6254[a]
	Test set	0.9022/0.9105[b]	0.5985/0.5920[b]	0.7435/0.9417[b]	0.6868/0.6107[b]

NOTE: [a] This value was obtained on the data set without the outliers identified by ECDF; [b] This value was obtained on the data points only involved in the AD determined by training and validation sets.

To the best of our knowledge, there is a lack of equivalent and available deep-learning models on the flammability-related properties that makes it difficult to compare the proposed MDNN with other existing models. Moreover, most existing models predict a single flammability-related property according to the manually selected molecule features. The proposed MDNN can correlate four flammability-related properties in a single model with molecular structures but not using pre-defined descriptors. Despite the large differences between the proposed model and existing single-task models, the proposed MDNN was still compared with the existing classical models (Section S6 of Supporting information †). The results show that the accuracy of the MDNN model is competitive to that of other types of classical models. Unlike some other studies those employ both experimental and prediction values, only the experimental values were used in this work. For the deep learning model, the smaller data sets could make the model performance less remarkable.

In addition, a multi-task model, although it is not a deep learning model, was built based on PLS[45] and Joback[63] group-contribution method and used to compare with the proposed MDNN. The multi-task PLS has been implemented in the previous study[45], but it requires all values of molecular features and target properties available for a compound at the same time; the finally employed data were less in PLS than those data correlated in the training of proposed MDNN.

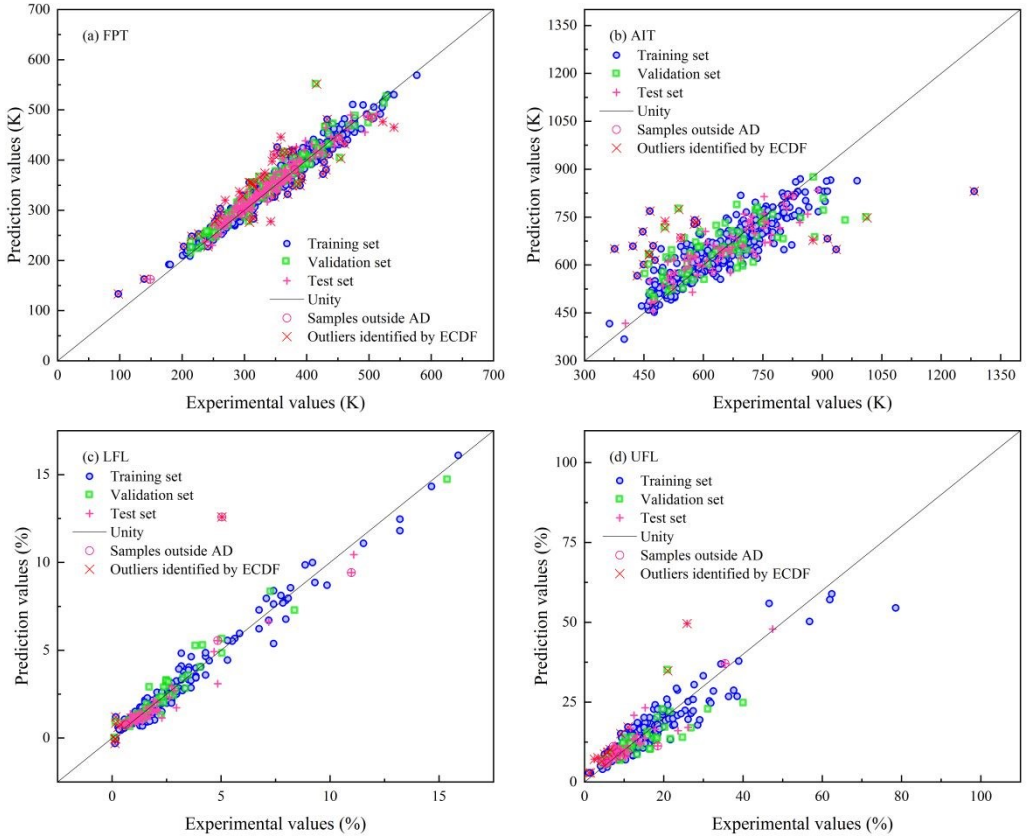


Fig. 5. The experimental and predicted values of four flammability-related properties for the training, validation and test sets obtained by the model (II): (a) FPT; (b) AIT; (c) LFL; (d) UFL.

The multi-task PLS model can also correlate four properties in a single model successfully, but its predictions are less accurate. Fig. 8 presents the deviations between the predicted values and experimental values obtained by the multi-task PLS. The PLS model provides lower values of Pearson correlation coefficient *r* than those given by the MDNN model. The PLS model performs undesirably on AIT and shows some lower errors on other three properties. This could be attributed to the data dispersion of properties. In particular, the combustion reaction is complicated and related to various factors, *e.g.*, chemical equilibrium, mass transfer, kinetics, *etc*. It is possible that uncertainty and inconsistent configurations in experiments may cause different measured values, and the impact of the gas composition is not frequently considered in experiments.[64] All these factors are not always available for each compound in the common-used databases, although the used data were carefully reviewed in DIPPR801 database.

The proposed multi-task learning strategy could be an approach to employ a unified molecular representation to correlate multiple properties in one model. We admit that it might be easier to obtain acceptable results *via* the previous single-task models specially designed for a unique property and with a particular representation of a molecule structure. However, these previous single-task models often describe molecular structures by various descriptors and their used data sets of properties could also be far different. As such, the ADs of various models are often different and there is a potential risk that various models may output different values on same compounds. The molecular representation can be unified in various models if the employed set of descriptors contains all the information of a chemical structure, however, this is practically impossible[65]. In this work, the proposed approach learns the molecular graphs directly, which aims to depict a molecule in 2D graphs as much as possible and unify the molecular representations for correlating various properties simultaneously, though all the precise information of

View Article Online
DOI: 10.1039/D1GC00331C

molecules cannot be still perfectly recorded. In other words, the selection and calculation of molecular descriptors can be eliminated in the modelling of QSPRs through the proposed strategy.

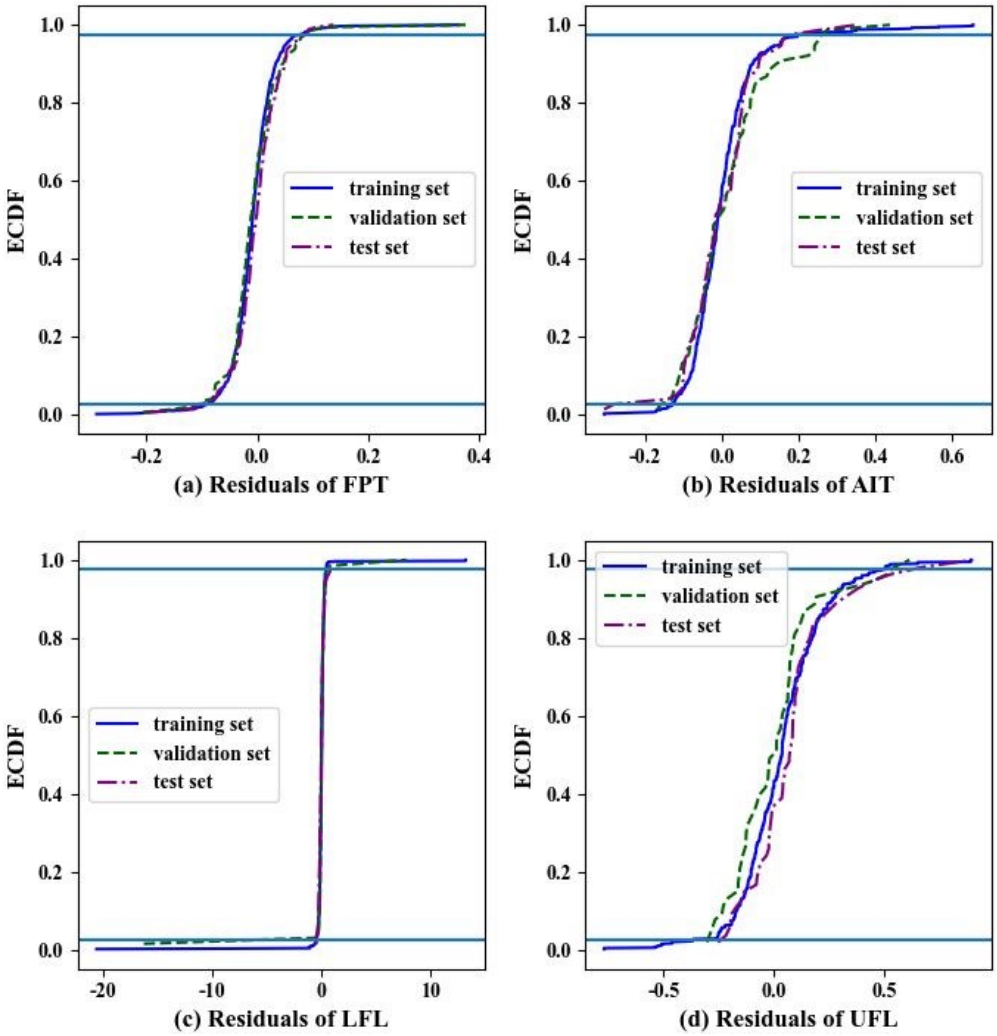


Fig. 6. The residual ECDFs for the training, validation, and test sets of: (a) FPT; (b) AIT; (c) LFL; (d) UFL, resulted by the model (Ⅰ)

4. Conclusions

To accelerate the process development at least during initial assessments and thus enable early go/no-go decisions in screening of the banned or restricted products based on the hazardous properties, a new methodology has been developed. This involves building QSPR models based on multi-task deep learning, involving data preparation, model training, outlier detection and AD determination. The methodology was successfully employed to correlate a new multi-task model for the simultaneous prediction of four flammability-related properties with a good accuracy. Compared with the multi-task learning technique of PLS, the proposed MDNN does not require that each molecule in the training set have a complete set of properties. Thus, the proposed MDNN can be applied on more samples and provides more accurate prediction. The proposed method can solve challenges in the descriptor-based QSPR modelling, *e.g.*, the development of a suitable descriptor for property correlation, the selection of descriptor and correlation analysis. As MDNN employs 2D structures (SMILES) rather than 3D, it is much easier to do massive screening of previously-unknown compounds without having to do 3D structural determination or prediction before property prediction. To avoid the risks of using the deep learning model, the outlier identification and AD determination were introduced into the evaluation of MDNN models. The residual ECDF was employed to identify these outliers. This study illustrates that it is feasible to observe the position of outliers in the latent chemical space with AD analysis based on PCA and the calculation of convex hulls. The results suggest that the ADs became narrower after outliers were excluded from training and validation sets. It can be also found that some compounds included in test sets appear outside the ADs. The exclusion of outliers could not necessarily improve the prediction ability of MDNN. The proposed method can identify whether the properties of a compound are estimated inside or outside the ADs of a deep learning model. Our strategy can open new avenues for modelling QSPRs with multiple-property outputs using the multi-task deep learning. This can be used as a promising tool for the data-driven virtual screening of green chemicals.

Conflicts of interest

There are no conflicts to declare.

View Article Online
DOI: 10.1039/D1GC00331C

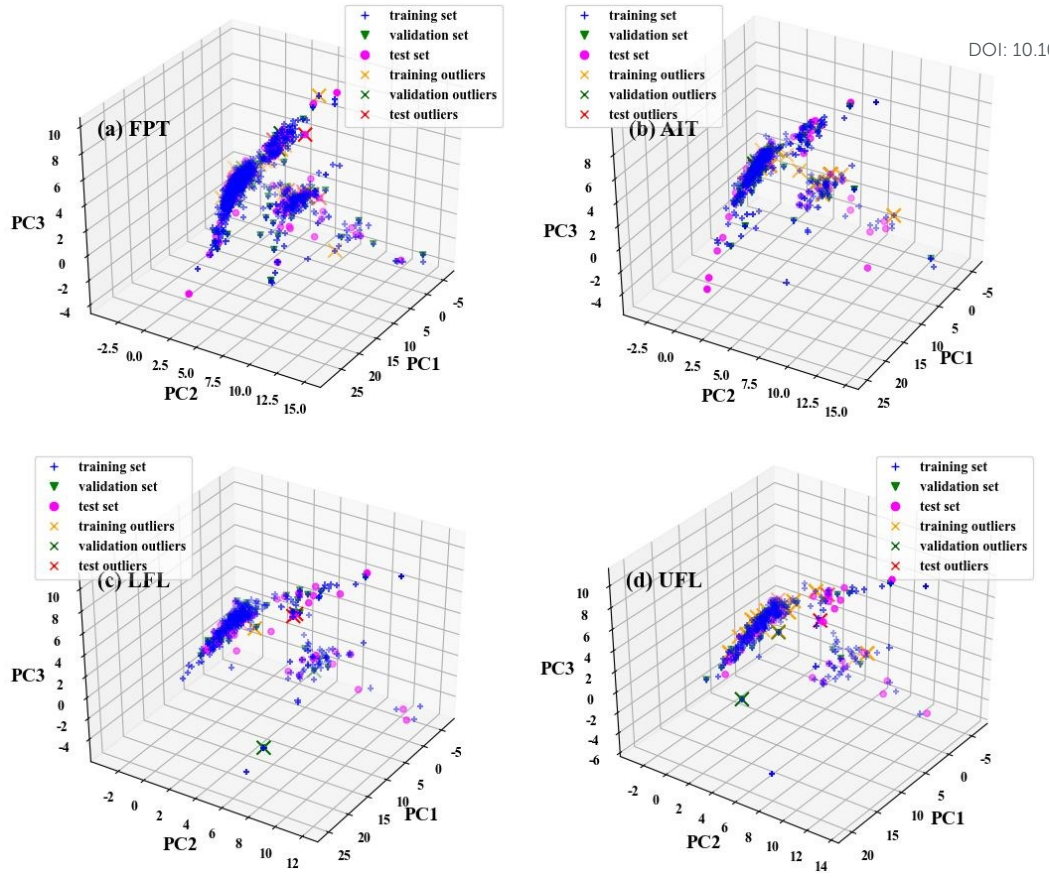


Fig. 7. The data points obtained with PCA of four flammability-related properties in the 3D space

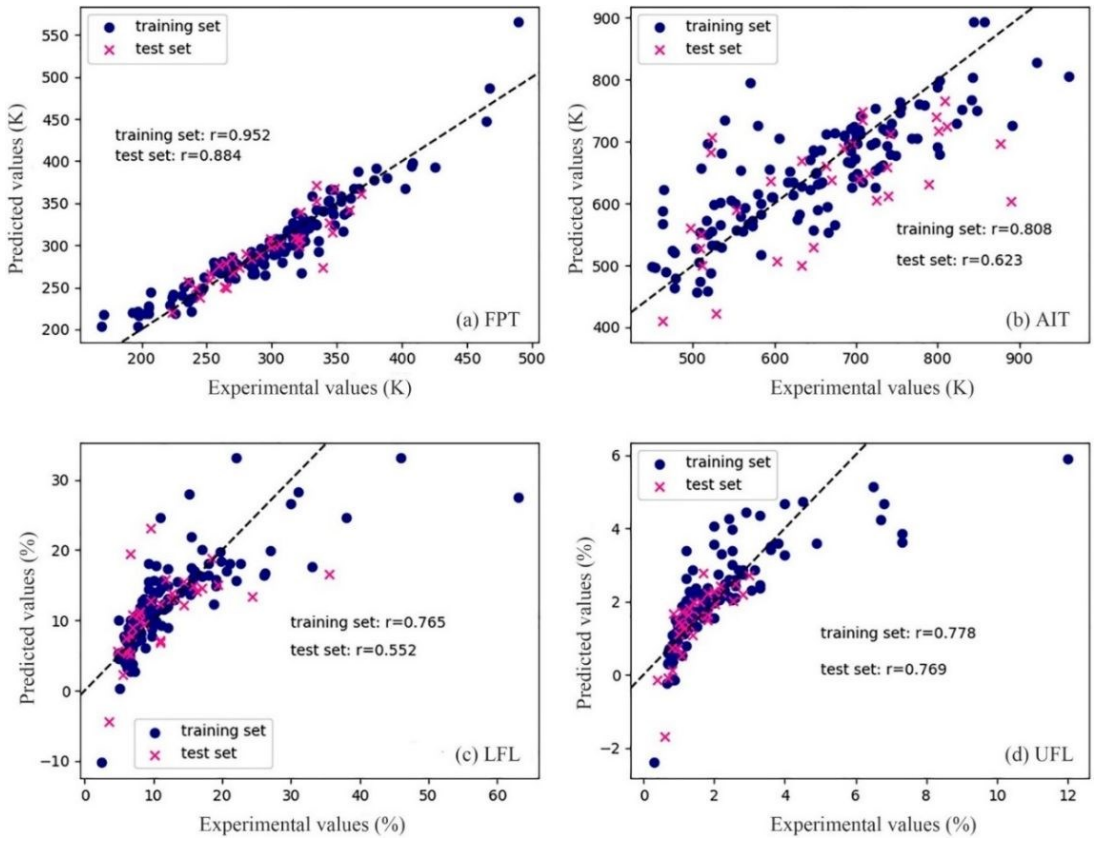


Fig. 8. The prediction versus the experimental values of the four flammability-related properties by the PLS-based multi-task model

View Article Online
DOI: 10.1039/D1GC00331C

Acknowledgement

We acknowledge the financial support provided by the National Natural Science Foundation of China (No. 21878028); the Beijing Hundreds of Leading Talents Training Project of Science and Technology (No. Z171100001117154); the Joint Supervision Scheme with the Chinese Mainland, Taiwan and Macao Universities - Other Chinese Mainland, Taiwan and Macao Universities (Grant No. SB2S).

Notes and references

1. F. A. Quintero, S. J. Patel, F. Muñoz and M. Sam Mannan, *Industrial & Engineering Chemistry Research*, 2012, **51**, 16101-16115.
2. A. S. Hukkerikar, B. Sarup, A. Ten Kate, J. Abildskov, G. Sin and R. Gani, *Fluid Phase Equilibria*, 2012, **321**, 25-43.
3. J. Frutiger, C. Marcarie, J. Abildskov and G. Sin, *Journal of Hazardous Materials*, 2016, **318**, 783-793.
4. F. Gharagheizi and R. F. Alamdari, *QSAR & Combinatorial Science*, 2008, **27**, 679-683.
5. S. J. Patel, D. Ng and M. S. Mannan, *industrial & Engineering Chemistry Research*, 2009, **48**, 7378-7387.
6. A. R. Katritzky, R. Petrukhin, R. Jain and M. Karelson, *Journal of chemical information and computer sciences*, 2001, **41**, 1521-1530.
7. F. Gharagheizi, *Journal of Hazardous Materials*, 2009, **169**, 217-220.
8. C. Nieto-Draghi, G. Fayet, B. Creton, X. Rozanska, P. Rotureau, J. C. de Hemptinne, P. Ungerer, B. Rousseau and C. Adamo, *Chemical Reviews*, 2015, **115**, 13093-13164.
9. Z. Jiao, H. U. Escobar-Hernandez, T. Parker and Q. Wang, *Process Safety and Environmental Protection*, 2019, **129**, 280-290.
10. F. Gharagheizi, M. H. Keshavarz and M. Sattari, *Journal of Thermal Analysis and Calorimetry*, 2012, **110**, 1005-1012.
11. F. Gharagheizi, *Journal of Hazardous Materials*, 2009, **167**, 507-510.
12. Y. Pan, J. Jiang, R. Wang, H. Cao and Y. Cui, *Industrial & Engineering Chemistry Research*, 2009, **48**, 5064-5069.
13. J. A. Lazzús, *Thermochimica Acta*, 2011, **512**, 150-156.
14. J. Frutiger, C. Marcarie, J. Abildskov and G. Sin, *Journal of Chemical & Engineering Data*, 2015, **61**, 602-613.
15. J. Frutiger, J. Abildskov and G. Sin, in *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, 2015, DOI: 10.1016/b978-0-444-63578-5.50079-7, pp. 503-508.
16. J. Jaworska, N. Nikolova-Jeliazkova and T. Aldenberg, *Alternatives to Laboratory Animals Atla*, 2005, **33**, 445-459.
17. A. S. Hukkerikar, S. Kalakul, B. Sarup, D. M. Young, G. Sin and R. Gani, *Journal of Chemical Information and Modeling*, 2012, **52**, 2823-2839.
18. J. C. Dearden, P. Rotureau and G. Fayet, *SAR and QSAR in Environmental Research*, 2013, **24**, 279-318.
19. S. Kim and K. H. Cho, *Bulletin of the Korean Chemical Society*, 2018, DOI: 10.1002/bkcs.11638.
20. A. Racz, D. Bajusz and K. Heberger, *Molecular Informatics*, 2019, **38**, e1800154.
21. T. Suzuki, K. Ohtaguchi and K. Koide, *Journal of Chemical Engineering of Japan*, 1991, **24**.
22. A. Alibakhshi, H. Mirshahvalad and S. Alibakhshi, *Process Safety and Environmental Protection*, 2017, **105**, 127-133.
23. T. A. Albahri and R. S. George, *Industrial & Engineering Chemistry Research*, 2003, **42**, 5708-5714.
24. T. Suzuki, *Fire and Materials*, 1994, **18**, 81-88.
25. A. A. Shimy, *Fire Technology*, 1970, **6**, 135-139.
26. T. A. Albahri, *Chemical Engineering Science*, 2003, **58**, 3629-3641.
27. F. Gharagheizi, *Journal of Hazardous Materials*, 2009, **170**, 595-604.
28. F. Gharagheizi, *Energy & Fuels*, 2008, **22**, 3037-3039.
29. M. Bagheri, M. Rajabi, M. Mirbagheri and M. Amin, *Journal of Loss Prevention in the Process Industries*, 2012, **25**, 373-382.
30. M. S. High and R. P. Danner, *Industrial & Engineering Chemistry Research*, 1987, **26**, 1395-1399.
31. J. R. Rowley, R. L. Rowley and W. V. Wilding, *Journal of Hazardous Materials*, 2011, **186**, 551-557.
32. G. B. Goh, N. O. Hodas and A. Vishnu, *Journal of Computational Chemistry*, 2017, **38**, 1291-1307.
33. Z. Cang and G. W. Wei, *PLoS Computational Biology*, 2017, **13**, e1005690.
34. S. K. Chakravarti and S. R. M. Alla, *Frontiers in Artificial Intelligence*, 2019, **2**, 17.
35. Z. Wang, Y. Su, W. Shen, S. Jin, J. H. Clark, J. Ren and X. Zhang, *Green Chemistry*, 2019, **21**, 4555-4565.
36. A. Mayr, G. Klambauer, T. Unterthiner and S. Hochreiter, *Frontiers in Environmental Science*, 2016, **3**, 80.
37. G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas and N. Baker, *arXiv*, 2017, **1706**, 06689.
38. G. B. Goh, N. O. Hodas, C. Siegel and A. Vishnu, *arXiv*, 2017, **1712**, 02034.
39. A. Lusci, G. Pollastri and P. Baldi, *Journal of Chemical Information and Modeling*, 2013, **53**, 1563-1575.
40. Y. Su, Z. Wang, S. Jin, W. Shen, J. Ren and M. R. Eden, *AIChE Journal*, 2019, **65**, e16678.
41. R. Gomez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernandez-Lobato, B. Sanchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Central Science*, 2018, **4**, 268-276.
42. B. Saha, S. Gupta, D. Phung and S. Venkatesh, *Knowledge and Information Systems*, 2015, **46**, 315-342.
43. R. A. Caruana, in *Machine Learning Proceedings 1993*, Morgan Kaufmann, 1993, pp. 41-48.
44. J. Wenzel, H. Matter and F. Schmidt, *Journal of Chemical Information and Modeling*, 2019, **59**, 1253-1268.
45. A. Varnek, C. Gaudin, G. Marcou, I. Baskin, A. K. Pandey and I. V. Tetko, *Journal of Chemical Information and Modeling*, 2009, **49**, 133-144.
46. K. S. Tai, R. Socher and C. D. Manning, 2015.
47. J.-L. Faulon, M. J. Collins and R. D. Carr, *Journal of Chemical Information and Modeling*, 2004, **44**, 427-436.
48. M. Tomas, S. Ilya, C. Kai, C. Greg and D. Jeffrey, presented in part at the Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Lake Tahoe, Nevada, 2013.
49. X. Glorot, A. Bordes and Y. Bengio, *Journal of Machine Learning Research*, 2011, **15**, 315-323.
50. V. Subramanian, *Deep learning with PyTorch*, Packt Publishing Ltd., Birmingham, 2018.
51. G. Landrum, *Journal*, 2019.
52. T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, *Computer Science*, 2013, 1-9.
53. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, Savannah, 2016.
54. D. Kingma and J. Ba, *Computer Science*, 2014, 1-15.
55. S. b. A. Design Institute for Physical Properties, *DIPPR Project 801 - Full Version*, Design Institute for Physical Property Research/AIChE, 2019.
56. S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang and E. E. Bolton, *Nucleic acids research*, 2019, **47**, D1102-d1109.
57. J. Godwin, Multi-Task Learning in Tensorflow (Part 1), (accessed 16/05, 2020).
58. R. Cipolla, Y. Gal and A. Kendall, 2018.
59. N. Reimers and I. Gurevych, 2017.
60. T. A. Albahri, *Process Safety and Environmental Protection*, 2015, **93**, 182-191.
61. Y. Pan, J. Jiang, R. Wang, H. Cao and J. Zhao, *Journal of Hazardous Materials*, 2008, **157**, 510-517.
62. G. Domenico, M. Giuseppe Felice, C. Marco, C. Angelo and N. Orazio, *International Journal of Quantitative Structure-Property Relationships (IJQSPR)*, 2016, **1**, 45-63.
63. K. G. Joback and R. C. Reid, *Chemical Engineering Communications*, 1987, **57**, 233-243.
64. A. Z. Mendiburu, J. A. de Carvalho and C. R. Coronado, *Fuel*, 2017, **188**, 212-222.
65. N. Nikolova-Jeliazkova and J. Jaworska, 2005, **33**, 461-470.