ACS CHEMICAL HEALTH & SAFETY

pubs.acs.org/acschas

Review

Machine Learning and Deep Learning in Chemical Health and Safety: A Systematic Review of Techniques and Applications

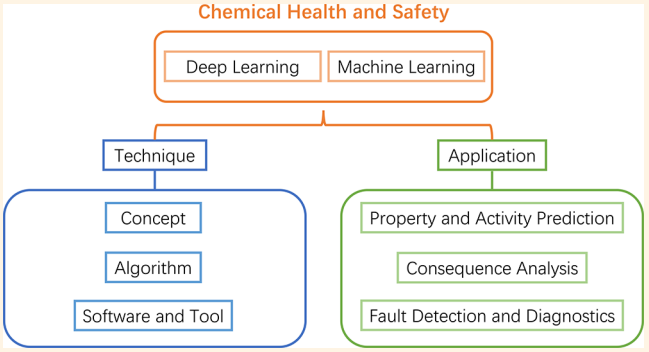
Zeren Jiao,[†] Pingfan Hu,[†] Hongfei Xu, and Qingsheng Wang*

Cite This: *ACS Chem. Health Saf.* 2020, 27, 316–334

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Machine learning (ML) and deep learning (DL) are a subset of artificial intelligence (AI) that can automatically learn from data and can perform tasks such as predictions and decision-making. Interdisciplinary studies combining ML/DL with chemical health and safety have demonstrated their unparalleled advantages in identifying trend and prediction assistance, which can greatly save manpower, material resources, and financial resources. In this Review, commonly used ML/DL tools and concepts as well as popular ML/DL algorithms are introduced and discussed. More than 100 papers have been categorized and summarized to present the current development of ML/DL-based research in the area of chemical health and safety. In addition, the limitation of current studies and prospects of ML/DL-based study are also discussed. This Review can serve as useful guidance for researchers who are interested in implementing ML/DL into chemical health and safety research and for readers who try to learn more information about novel ML/DL techniques and applications.



KEYWORDS: *machine learning, deep learning, artificial intelligence, chemical health, process safety*

1. INTRODUCTION

Machine learning (ML) is an interdisciplinary area, involving probability theory, statistics, approximation theory, convex analysis, algorithm complexity theory, and other disciplines. It is the core subset of artificial intelligence (AI). Machine learning algorithms can build a mathematical model based on training data for predictions or decisions without being explicitly programmed to do so. Dating back to the 17th century, Bayesian and Laplace's derivations of least-squares and Markov chains constituted the tools and foundations widely used in machine learning.[1] The term "machine learning" was first proposed in 1959 by Arthur Samuel.[2] Since then, the machine learning algorithm has developed tremendously, and has been widely applied in various aspects of scientific research and our daily lives such as data mining,[3] computer vision,[4] natural language processing,[5] biometric recognition,[6] medical diagnosis,[7] detection of credit card fraud,[8] stock market analysis,[9] speech and handwriting recognition,[10] strategy games,[11] and robotics.[12]

Deep learning (DL) is a relatively new branch direction in the field of ML. It is an algorithm that uses artificial neural networks as the architecture to characterize and learn data. The concept of DL originates from the research of artificial neural networks, and a multilayer perceptron with multiple hidden layers is a deep learning structure.[13] DL forms a more abstract high-level representation attribute category or featured by combining low-level features to discover distributed feature representations of data. There have been several deep learning frameworks, such as the deep neural network (DNN), convolutional neural network (CNN), and recurrent neural network (RNN).

The applications of ML algorithms in chemical health and safety study can date back to the mid-1990s.[14] Some research used basic ML algorithms in toxicity classification and prediction studies. For other fields such as hazardous property prediction and consequence analysis, the implementation of ML/DL algorithms did not emerge until the late 2000s.[15,16] Chemical health and safety, although an important field of study, has always been in a backward position in interdisciplinary research with applied machine learning. This is because at the early development stage of ML/DL, the algorithm was relatively primitive, and its excellent prediction accuracy and capability were not widely verified and proven. Second, due to the lack of relatively simple and easy-to-use toolkits and the high skill requirement of algorithms and programming, the

Received: August 1, 2020
Published: October 18, 2020

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316–334

applications of ML/DL algorithms in chemical health and safety research are limited. As a result, studies implementing machine learning have been relatively rare in the field of chemical health and safety in the late 20th century and the first decade of the 21st century.

However, with the rapid advancement of AI and computer science in the past ten years, the importance of ML/DL and their unparalleled advantages over traditional statistical methods and labor-intensive works have drawn much more attention and hence have developed significantly recently. There is also a growing interest in expanding the application of ML/DL in the research field of chemical health and safety. However, there is currently a lack of a comprehensive review which can give the readers an introduction to ML/DL and summarize the current development of ML/DL applications in chemical health and safety research.

In this Review, commonly used ML/DL tools as well as popular ML/DL algorithms for the implementation of ML/DL in chemical health and safety research are introduced. More than 100 papers have been categorized and summarized in this Review to present the current development. In addition, the limitations of current studies and future directions of ML/DL-based study are also discussed. This Review can serve as useful guidance for researchers who are interested in implementing ML/DL into chemical health and safety research and for readers who try to find more information about novel ML/DL tools and algorithms.

2. CONCEPT AND ALGORITHM

2.1. Supervised Learning and Unsupervised Learning. *2.1.1. Supervised Learning.* Supervised learning learns a function from a given training data set. When new data (validation/test data) comes, it can predict the results based on the function. The training set requirements for supervised learning include inputs (features) and outputs (targets). The targets in the training set are already labeled (with specific experimental/simulation values). Common supervised learning algorithms include regression algorithms and classification algorithms. While some algorithms are only capable of classification analysis (e.g., linear discrimination analysis, naïve Bayes classification, etc.), most of them (e.g., *k*-nearest neighbor, random forest, etc.) are able to conduct both classification analysis and regression analysis.[17,18]

2.1.2. Unsupervised Learning. The difference between supervised learning and unsupervised learning is whether the target of the training set is labeled. They all have training sets, and they all have inputs and outputs. Compared with unsupervised learning, the training set has no artificially labeled results. Common unsupervised learning algorithms can be used for clustering.[17,18] There are also semisupervised learning, which is between supervised learning and unsupervised learning, and reinforcement learning. The algorithm of semisupervised learning gradually adjusts its behavior as the environment changes.

2.2. Machine Learning Algorithm. *2.2.1. Naïve Bayes.* The naïve Bayes (NB) classifier is a simple probabilistic machine learning model for classification tasks. Naïve Bayes applies Bayes' theorem with the "native" assumption that the value of a feature is independent of the value of any other features. According to Bayes' theorem, given a class variable y and its dependent feature vector $x = (x_1, ..., x_n)$, the following relationship stands.[19]

$$P(y|x_1, ..., x_n) = \frac{P(y)\, P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)}$$

Naïve conditional independence assumptions imply the following.

$$P(x_i|y, x_1, ..., x_i, x_{i+1}, ..., x_n) = P(x_i|y)$$

Thus

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i+1}^{n} P(x_i|y)}{P(x_1, ..., x_n)}$$

In addition to the model, the naïve Bayes classifier requires a decision rule. A common one is *Maximum A Posteriori* (MAP), which picks the most probable hypotheses. Using MAP estimation, we can estimate $P(y)$, which is the relative frequency of class y in the training set, and $P(x_i|y)$. In order to perform parameter estimation, various assumptions regarding the distribution of $P(x_i|y)$ can be made.[19]

Naïve Bayes classifiers only require a small amount of data for parameter estimation and have worked quite well in real-world situations, even though they apply oversimplified assumptions.

2.2.2. Linear Discriminant Analysis. Linear discriminant analysis (LDA) was first developed in 1936 by R. A. Fisher which can be used as a linear classifier, dimension reduction before classification, or virtualization.[20] LDA was established on some simple assumptions. First, the data is assumed to be Gaussian distributed. Second, the variance for each variable is the same. Following Bayesian rule, the data x to class j can be classified by achieving the highest likelihood among all K classes for $i = 1$ to K. Using the maximum likelihood rule and Bayesian rule, the discriminant function can be established, which shows how likely data is from each class. By assuming equal covariance among K classes, a linear function in x can be established, i.e., linear discriminant analysis, which means the decision boundary between any pair of classes is also a linear function in x. In addition to classification, LDA can also be used for dimension reduction. By identifying a smaller number of meaningful dimensions, LDA can be used to visualize the high-dimension problem.

Although LDA is simple, with a linear decision boundary and the ability of dimension reduction, LDA still has several limitations. The linear decision boundary may not be enough for class separation. It may use too many parameters for a high-dimensional setting.

2.2.3. k-Nearest Neighbor. *k*-nearest neighbor (*k*-NN) is one of the simplest machine learning methods proposed by Thomas Cover which can be applied in both classification and regression problems.[21] An example of the *k*-NN algorithm is shown in Figure 1. For *k*-NN classification, the classification of the object (green square) is determined by the majority vote of k nearest neighbors, which is yellow when $k = 3$ and red when $k = 5$. For *k*-NN regression, the predicted value is the average of the kth nearest neighbor value which is $\frac{a_1 + a_2 + a_3}{3}$ when $k = 3$ and $\frac{a_1 + ... + a_5}{5}$ when $k = 5$. This method is highly efficient in the low-dimensional problem, which has fewer predictors. However, it may suffer from the increasing distance between neighbors when the dimension increases, which is called "curse of dimensionality". It is also worth noting that since the prediction highly depends on the distance, normalization of the data set is recommended for result improvement.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

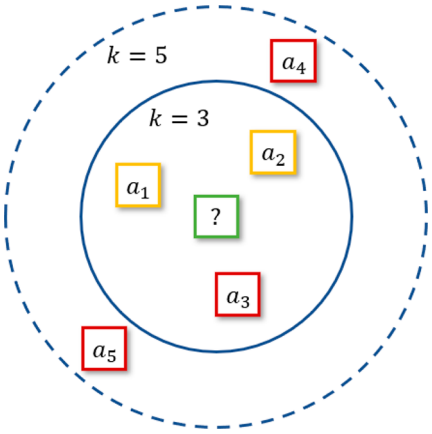


Figure 1. Example of the *k*-NN algorithm.

2.2.4. Support Vector Machine. Support Vector Machine (SVM) is a machine learning method which can be applied to classification and regression problems in supervised learning. It is also one of the most used machine learning algorithms in chemical health and safety-related research. The algorithm is proposed by Vapnik in AT&T Bell Laboratories. The SVM is originally used for a classification program. A diagram of the SVM classification is shown in Figure 2. The fundamental

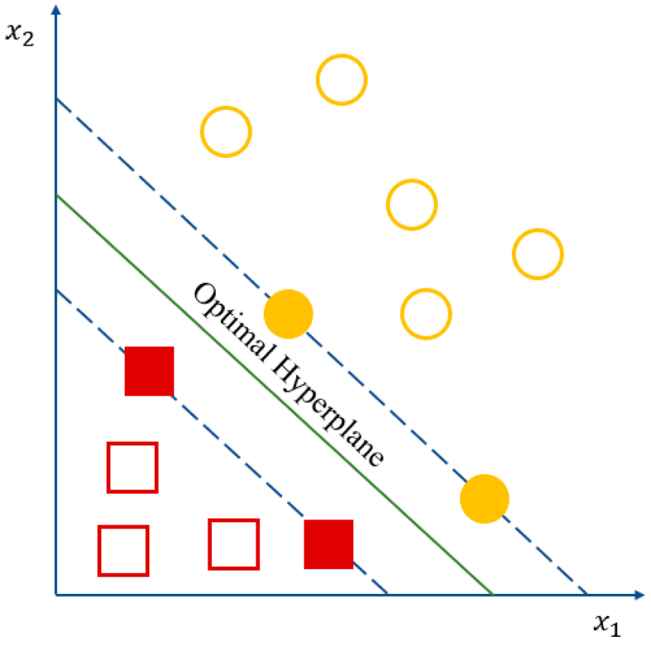


Figure 2. An example of the SVM linear classification algorithm.

principle of SVM to is to find the optimal hyperplane in data space so that the data point can be separated into categories and divided by a widest possible gap. New objects which need to be predicted are then mapped into the data space, and the classification is determined based on the side of the gap on which they fall. The linear classification can also be expanded toward nonlinear classification by using kernel functions to map inputs into high-dimensional feature spaces.

With the introduction of the ε-insensitive loss function, the SVM application is extended to the regression problem (support vector regression, SVR). The regression application of SVM was first proposed by Drucker et al.[22] The objective of SVM-based regression is to minimize the coefficients the l_2-norm of the coefficient vector with constraint shown below:

$$\begin{cases} \text{minimize: } \min \dfrac{1}{2} \| w \|^2 \\ \text{constraint: } |y_i - \langle w, x_i \rangle - b| \le \varepsilon \end{cases}$$

The SVM model in regression relies only on the subset of the training set since the cost function ignores data points close to the prediction. The kernel function is the critical component in the SVM method which can introduce nonlinearities into the model. For the regression problem, the radical kernel function is more common than the linear and polynomial kernel. Furthermore, the SVM-based methodology is also expanded to be applied in unsupervised learning which is called support-vector clustering (SVC).[23]

The reason why SVM is more widely applied in research, apart from its prediction accuracy, is because of the wide distribution of LIBSVM. LIBSVM is developed at National Taiwan University which implements the sequential minimal optimization (SMO) algorithm for kernelized SVM classifications and regression applications.[24] It has also been widely reused in different open-source machine learning toolkits such as KNIME and Orange, as well as programming packages such as Scikit-learn of Python[25] and the *e1071* package of R.[26]

2.2.5. Random Forest. Random forest (RF), or random decision forest, is a tree-based ensemble learning method for classification and regression that improves on the bagged tree method. The diagram of the random forest algorithm is shown in Figure 3. Instead of establishing a single decision tree, RF constructs a multitude of decision trees with randomly selected predictors to minimize the correlation among predictors. This method can significantly lower the variance and correct the overfitting problem posed by the decision tree when the "forest" is averaged.

The RF algorithm was first developed by Tin Kam Ho using a random subspace method and further expanded by Leo Breiman.[27,28] The RF algorithm can also be expanded toward unsupervised learning by constructing a random forest predictor that distinguishes the "observed" data from suitably generated synthetic data.[28] The kernel method introduced in SVM can also be combined with RF (KeRF), which can make the RF algorithm more interpretable and easier to analyze.[29]

2.2.6. Gradient Boosting. Gradient Boosting (GB) is a machine learning method for regression and classification problems in the form of an ensemble of weak prediction models. This method builds models in a stage-wise fashion and generalizes models by allowing optimization of an arbitrary differentiable loss function.[30] The idea of gradient boosting originated from the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function,[31] which was later explicitly developed by Friedman.[32] The GB algorithms can be viewed as an optimization of the cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This view of GB has led to the development of boosting algorithms in many areas beyond regression and classification problems.

Among various applications of the GB technique, the gradient tree boosting is the most widely used derivative algorithm. The diagram of gradient tree boosting is shown in Figure 4. Instead of fitting the tree to the entire data set like the RF method, the GB-based method develops the tree

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

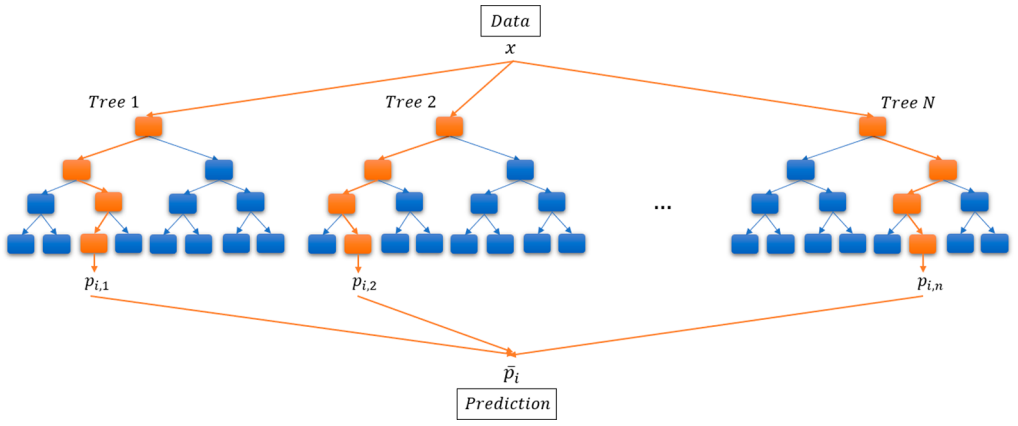


Figure 3. Example of random forest algorithm.

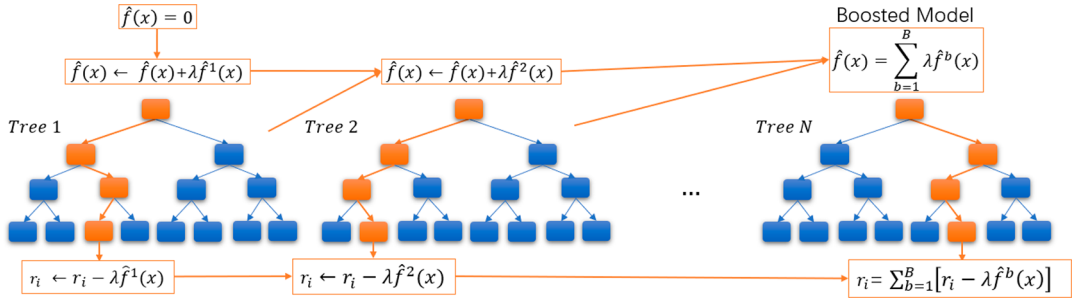


Figure 4. Example of the gradient boosting tree algorithm. Reproduced with permission from ref 133. Copyright 2020 American Chemical Society.

sequentially by fitting the tree to the residuals of the previous decision tree. Instead of fitting the data hard, the boosting tree method learns slowly by only fitting the predictors to the updated residuals from the previous tree.[18] Besides, when applying the GB algorithm, fitting the training set too closely can lead to the degradation of a model's generalization ability. This problem can be solved using regularization techniques by constraining the fitting procedure.

Many tools can be implemented in GB-based machine learning model development. Among those, *Xgboost* is proven to have superior performance to other machine learning algorithms.[33] *Xgboost* is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the gradient boosting framework. *Xgboost* can provide a parallel tree boosting that solves many data science problems in a fast and accurate way.[30]

2.2.7. k-Mean Clustering. *k*-mean clustering is an unsupervised machine learning method that aims to partition the data set into *k* clusters with the nearest mean.[34] The *k*-mean clustering starts with initialization, which is the assignment of observations to start the clustering process and to update the step which calculates the centroids (mean) of the cluster and continues to iterate until the convergence is reached. There are two main initialization methods currently used in machine learning clustering, the Forgy method and random partition method.[35] The Forgy method randomly chooses *k* data points in the data set as the cluster mean and proceeds to the update step. The random partition on the other hand randomly assigns a cluster for each data point at initialization. The Forgy method tends to spread the initial means out, while random partition places all of them close to the center of the data set.

The result of *k*-mean clustering heavily depends on the *k* value selection. An inappropriate *k* selection can lead to a poor clustering result. In order to determine the optimal *k* of the data set, the elbow method is often used to intuitively select the *k* value of the data set. The elbow method is a heuristic method that plots the explained variation under different *k* values. The optimal *k* is the value that meets certain criteria or has minimal model improvement.

2.2.8. Hierarchical Clustering. Hierarchical clustering analysis (HCA) is a clustering method for unsupervised learning which builds a hierarchy of the classified clusters. The HCA was first proposed by Johnson in 1967.[36] The HCA has two main clustering methods, the bottom-up (agglomerative) method and top-down (divisive) method. The merge and split of the data set are determined by greedy search, and the result can be presented via a dendrogram. An example of HCA and the associated dendrogram is shown in Figure 5. HCA builds a cluster tree to represent data, where each node links to two or more successor nodes. The clustering is determined by the cluster dissimilarity, which is determined by both metric (distance measure by certain approach) and linkage criteria.

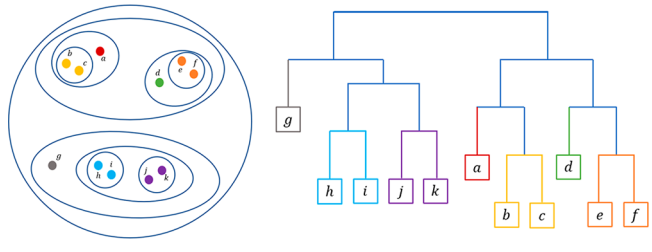


Figure 5. Example of HCA and dendrogram.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

The groups are nested and organized as a tree, which ideally ends up as a meaningful classification scheme.

For applications, most of the software contains a hierarchical clustering module like MATLAB, SAS, and Mathematica. For the open-source toolbox, R and Python also have various packages that provide functions for HCA.

2.2.9. Density-Based Spatial Clustering of Applications with Noise. Density-based spatial clustering of applications with noise (DBSCAN) is a clustering algorithm proposed by Ester et al.[37] DBSCAN ignores the low-density area and treats them as noise. There are two core parameters of the DBSCAN algorithm, *eps* (ε) and *minPts*. *eps* is the distance neighbor determination criteria, which is the maximum distance that two data points can be considered a neighbor. *minPts* is the minimum data point number that can define a cluster. The DBSCAN method is to find the points in the neighborhood under *eps* criteria of every point and to identify the core points with more than *minPts* neighbors.

Compared to *k*-mean clustering, DBSCAN can take an irregular shape, and it also works well with noisy data sets. *k*-mean clustering, due to its dependence on the *k* value, relies on techniques such as the elbow method to choose the optimal value. The *k*-mean method does not work well with outliers and high-dimensional problems, while DBSCAN tends to handle it well.[38]

2.2.10. Single Hidden Layer Neural Network. The original work on neural networks was published by Warren McCulloch and Walter Pitts in 1943.[39] They introduced the McCulloch–Pitts neural model, also known as the linear threshold gate. As the first computational model of a neuron, the McCulloch–Pitts neural model is very simplistic, which only generates a binary output. The weights and threshold require hand-tuning. In the 1950s, the perceptron became the first model with the capability to automatically learn the optimal weight coefficients, allowing the training of a single neuron.[40] With the help of the backpropagation algorithm, people started to train a neural network with one or two hidden layers.[41]

A single hidden layer neural network consists of 3 layers: input layer, hidden layer, and output layer. In the neural network that is trained with supervised learning, the training set contains values of the inputs x and the target outputs y. The hidden layer refers to the fact that, in a training set, the true values for these nodes are not observed. As shown in Figure 6, a notation for the values of the input features is $a^{[0]}$, where the term a stands for activation. It refers to the values that different layers of the neural network are passing on to the subsequent layers. After the input layer passes on the values x to the hidden layer, the hidden layer in turn generates some sets of activations, $a^{[1]}$. Finally, the output layer generates some value $a^{[2]}$, which is a real number that equals $\hat{y}$. The hidden layer and output layer have associated with the parameters w and b. In order to compute the outputs (a) of the neural network, which is a sigmoid function of z ($\sigma(z)$), it is similar to operate repeated logistic regression. The calculations are shown in eqs 1–4. Besides the sigmoid function, other activation functions could be used to compute the hidden layer values. In modern neural networks, the default recommendation is to use hyperbolic tangent (tanh) or the rectified linear unit (ReLU).

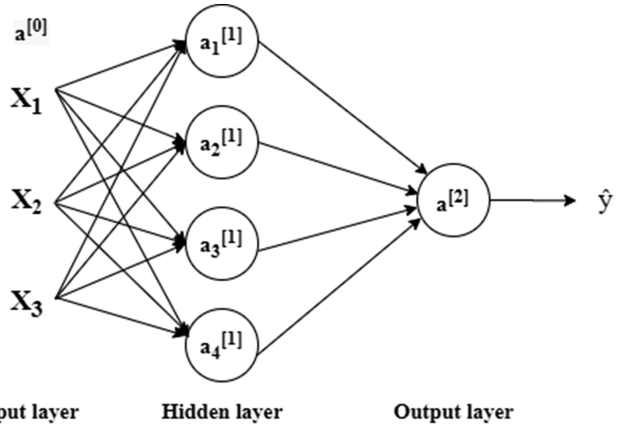


Figure 6. Structure of a single hidden layer neural network.

$$z_1^{[1]} = W_1^{[1]T}x + b_1^{[1]}, \quad a_1^{[1]} = \sigma(z_1^{[1]}) \tag{1}$$

$$z_2^{[1]} = W_2^{[1]T}x + b_2^{[1]}, \quad a_2^{[1]} = \sigma(z_2^{[1]}) \tag{2}$$

$$z_3^{[1]} = W_3^{[1]T}x + b_3^{[1]}, \quad a_3^{[1]} = \sigma(z_3^{[1]}) \tag{3}$$

$$z_4^{[1]} = W_4^{[1]T}x + b_4^{[1]}, \quad a_4^{[1]} = \sigma(z_3^{[1]}) \tag{4}$$

2.2.11. Deep Neural Network. Over the past ten years, the machine learning community has realized that some cases could only be learned by a deep neural network (DNN) rather than the single hidden layer neural network.[42] The DNN with multiple hidden layers can have the earlier layers learn about low-level simpler features and then have the later deeper layers detect more complex features. Compared with the shallower neural network, the DNN requires significantly fewer hidden units to compute. Although for any given problem it can be hard to predict in advance exactly how deep a neural network should be, the number of hidden layers can be treated as a hyperparameter and be evaluated by hold-out cross-validation data. The DNN relies on both forward propagation and backpropagation. The forward propagation allows the input to provide the initial information and to propagate up to the subsequent layers, while the backpropagation allows the information flow backward from the cost to compute the gradient more efficiently. Figure 7 summarizes the calculation of a DNN with four hidden layers under both forward and backward propagation. Figure 8 shows the structures of some typical neural networks that are useful to the chemical engineering field.

2.3. Cross-Validation. Cross-validation can be categorized into two types, exhaustive and nonexhaustive cross-validation. Exhaustive cross-validation involves learning and testing on all possible ways to divide the original sample into training and testing data sets.[43] On the contrary, nonexhaustive cross-validation methods do not compute for all possible combinations of the original data and thus are an approximation of leave-p-out cross-validation. One example of nonexhaustive cross-validation methods is the *k*-fold cross-validation method. As shown in Figure 9, the data set is first divided into k equally sized parts. The *k*-fold statistical assessment value is calculated by average through all developed models. Since the model is developed and validated based on the whole data set, the performance evaluation can be more equal and stabilized due to the elimination of uncertainty from data separation. There are also variants of *k*-fold cross-validation like the nested cross-validation method.[44]

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316–334

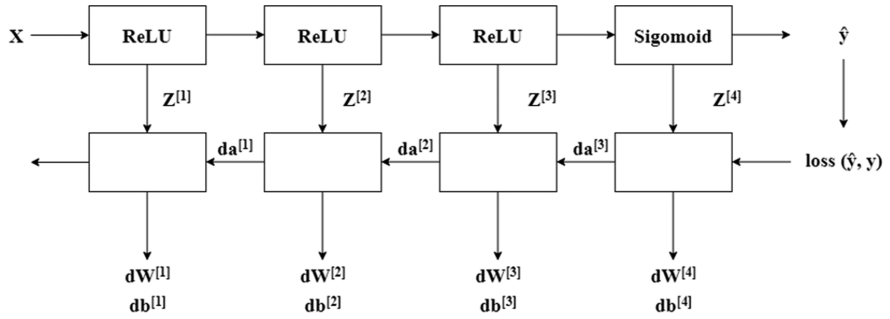


Figure 7. Calculation of a deep neural network with four hidden layers.

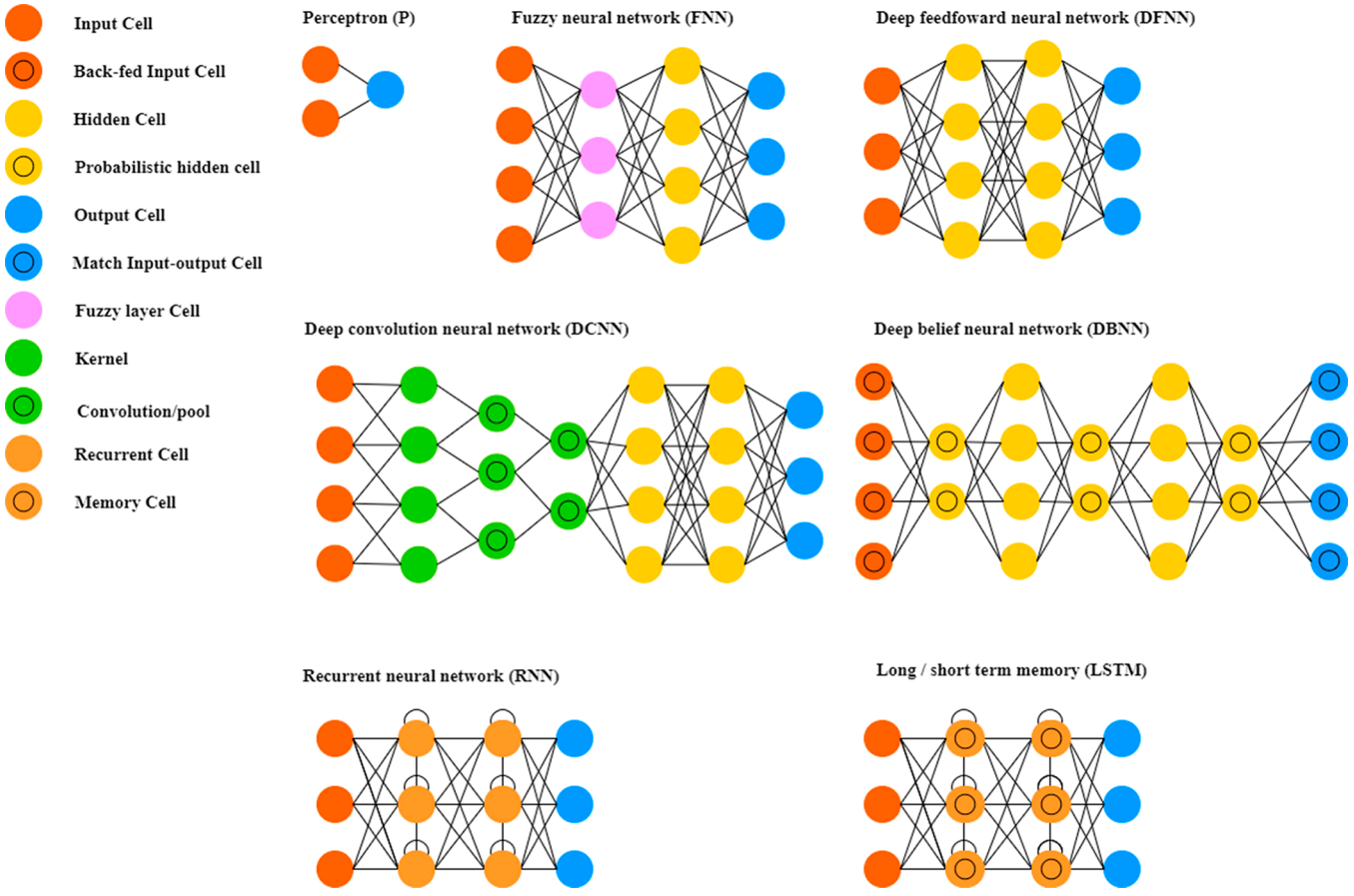


Figure 8. Structures of some typical neural networks.

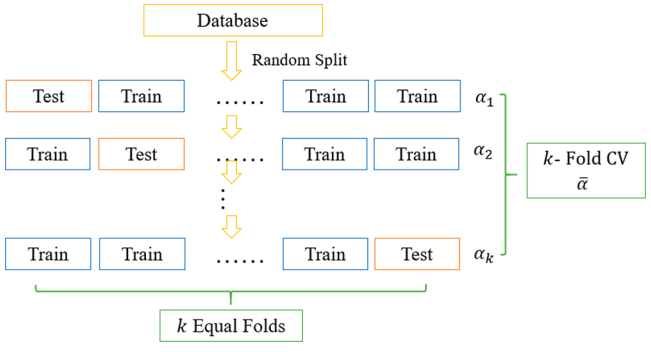


Figure 9. Main architecture of *k*-fold cross-validation. Reproduced with permission from ref 61. Copyright 2020 Elsevier.

3. SOFTWARE AND TOOLS

3.1. R. R is a language and operating environment for statistical analysis and machine learning. R is a branch of the S language that was created around 1980 and is widely used in the statistical field. R is free and open-source software belonging to the GNU system, and it is an excellent tool for statistical calculation and machine learning model construction.[45]

The use of the R language is largely aided by various R packages. To some extent, R packages are plug-ins for R. Different plug-ins meet different needs. CRAN (Comprehensive R Archive Network) has already included more than 16 000 packages of various types including many popular ML/DL packages like randomForest,[46] Xgboost,[47] e1071,[48] TensorFlow,[49] etc.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

3.2. Python. Python is a cross-platform, general-purpose programming language which was created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. It is a high-level scripting language that combines explanatory, compliable, interactive, and object-oriented features. With the continuous update of the version and the addition of new language features, the more it is used for the development of independent and large-scale projects. It is also the go-to tool for beginners as well as professionals to learn and use ML/DL algorithms.[50]

Compared to proprietary software like MATLAB, using open-source programming languages like Python for ML/DL model development has some important advantages: First, MATLAB is an expensive proprietary software. Python on the other hand is completely free, and many open-source ML/DL and scientific computing libraries provide Python calling interfaces. In addition to some highly specialized toolboxes of MATLAB that cannot be replaced, most of the commonly used functions of MATLAB can be found in Python. Users can install Python and most of its extension libraries on any computer for free, and Python also has the most state-of-the-art ML/DL library available that can easily complete various advanced tasks and achieve a superior performance. In addition, compared with MATLAB, Python is a programming language that is easier to learn and more rigorous which can make composed code easier for users to write, read, and maintain.

3.3. Scikit-learn. Scikit-learn is a Python library integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. It originated from a project called scikits.learn started by a Google Summer of Code project. This package focuses on bringing machine learning to nonspecialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and application programming interface (API) consistency. It features various classification, regression, and clustering algorithms including support vector machines (LIBSVM integrated), random forests, gradient boosting (Xgboost integrated), *k*-means, and DBSCAN and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Since its wide encapsulation of various ML algorithms, it is the most popular and one-stop choice of the library for ML model construction.[25]

3.4. TensorFlow. TensorFlow is an open-source machine learning platform for both R and Python. It works by building a graph of defined computations. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.[49] The data set can be created and preprocessed by tf.data and TFRecord. The eager execution mode with the API framework, Keras, can be combined with the visualization tool, TensorBoard, to build and debug models effectively. Then, the model can be trained on the CPU/GPU. It is a powerful tool for different neural networks such as the convolutional neural network (CNN), recurrent neural network (RNN), and deep reinforcement learning (DRL).

4. APPLICATION

4.1. Property and Activity (Toxicity) Prediction. Accurate chemical property/toxicity values are extremely important to process safety, industrial hygiene, and novel chemical development. Experimental measurement is the most commonly used method to determine property/toxicity values.[51] However, the experimental setup of the property measurement is expensive, and most of the chemicals are highly flammable and toxic, for which it is extremely dangerous to conduct the experiment. For chemical mixture property/toxicity predictions, mixtures have various combination patterns, making it a highly time-consuming task to measure all mixture combinations.[52]

Quantitative structure−activity/property relationship analysis (QSAR/QSPR) involves regression and classification models and is widely used in biological and pharmaceutical science and engineering.[53,54] It also has been intensively used in chemical health and safety research recently due to its high prediction accuracy and reliability.[55−57] In addition, it is the area in which machine learning tools have been most extensively applied to assist model development. This is because QSAR/QSPR has a well-developed data pipeline which can greatly facilitate the development of machine learning-based QSAR/QSPR studies.

The standard procedure of QSPR/QSAR model development is shown in Figure 10. The first step is to establish the

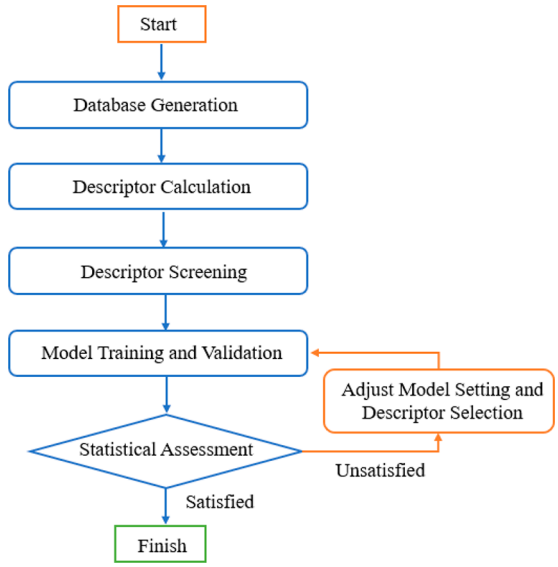


Figure 10. Procedure of QSAR/QSPR model development. Reproduced with permission from ref 61. Copyright 2020 Elsevier.

data set. Some guidelines should be followed when gathering the data. First, the data set of a valid machine learning-based model should have at least 50 data points. When gathering data, the experimental conditions of the property/toxicity measurement should be carefully examined to ensure only experiments conducted under similar conditions are included. After the generation of the database, the predictors should be gathered or calculated. In a QSAR/QSPR study, the predictors/features are physicochemical properties or theoretical molecular descriptors. After the descriptor calculation, there are thousands of molecular descriptors that can be categorized into 0-D to 4-D descriptors. However, the QSAR/QSPR model can only have a limited number of features/predictors to avoid the overfitting problem. In most QSAR/QSPR studies, the features/predictors are mostly limited to 4−8. It also depends on the size of the data set. For example, a

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

Table 1. Summary of Machine Learning-Based QSPR Studies

name	year	validation	method	property	software	package/module	ref.
Bagheri et al.	2012	validation set	ANN	FP/AIT	MATLAB	Neural Network Toolbox	62
Borhani et al.	2016	validation set	ANN	AIT	MATLAB	Neural Network Toolbox	69
Gharagheizi	2011	validation set	ANN	AIT	MATLAB	Neural Network Toolbox	70
Jiao et al.	2016	*k*-fold	ANN	FP	MATLAB	Neural Network Toolbox	71
Jiao et al.	2019	*k*-fold	SVM/*k*-NN/RF/GB	mixture LFL	R	e1071/randomForest/FNN	52
Jiao et al.	2020	*k*-fold	GB	mixture LFL/UFL/AIT/FP	R	Xgboost	61
Jin et al.	2020	validation set	SVM	AIT	MATLAB	LIBSVM	72
Khajeh and Modarress	2010	validation set	ANN	FP	MATLAB	Fuzzy Logic Toolbox	63
Mallakpour et al.	2013	validation set	ANN	polymer temperature of five percent of decomposition (T5)	MATLAB	Neural Network Toolbox	73
Mallakpour et al.	2014	validation set	SVM	polymer temperature of five percent of decomposition (T10)	MATLAB	LIBSVM	74
Mirshahvalad et al.	2018	validation set	ANN	FP	MATLAB	Neural Network Toolbox	75
Pan et al.	2008	validation set	SVM	AIT	MATLAB	LIBSVM	15
Pan et al.	2009	validation set	SVM	LFL	MATLAB	LIBSVM	16
Saldana et al.	2011	*k*-fold/LOOCV	ANN/SVM	FP/cetane number	Python		76
Saldana et al.	2013	validation set	SVM	FP	Python		77
Venkatraman et al.	2016	*k*-fold	RF	thermal decomposition temperatures	R	VSURF/randomForest	66
Wang et al.	2017	validation set	SVM	MIE	MATLAB	LIBSVM	78
Wang et al.	2017	validation set	SVM	SADT	MATLAB	LIBSVM	56
Wang et al.	2019	validation set	SVM	mixture UFL	MATLAB	LIBSVM	55
Wang et al.	2020	validation set	SVM	mixture superheat limit temperatures	MATLAB	LIBSVM	79
Yuan et al.	2019	validation set	SVM/*k*-NN/RF	UFL	Python	Scikit-learn	64
Yuan et al.	2019	validation set	*k*-mean/HCA	liquid flammability ratings	Python	Scikit-learn	68
Zhang et al.	2020	validation set	k-mean/HCA	chemical logistic warehouses hazard index	Python	Scikit-learn	67
Zhou et al.	2017	validation set	SVM	gas−liquid critical temperatures	MATLAB	LIBSVM	57

smaller descriptor set should be selected if the data set is relatively small.[58]

The next step is model training and validation. For machine learning-based model development, the parameters of the model need to be carefully tuned to reach optimal performance. A comparison of different algorithms is also recommended for selecting the algorithm which can reach the highest accuracy. In addition, per Principle 3 of Organization for Economic Co-operation and Development (OECD), the applicability domain should be defined for developed QSAR/QSPR models to determine whether the model's assumptions are met and to which the QSAR/QSPR model can be reliably applied.

For property predictions, the main target properties are lower flammability limit (LFL), upper flammability limit (UFL), autoignition temperature (AIT), and flash point (FP). There are also other properties that have been investigated such as minimum ignition energy (MIE) and self-accelerating decomposition temperature (SADT), but only limited studies are available due to the amount of data available.

A summary of the machine learning-based QSPR model for hazardous property prediction is shown in Table 1. A QSPR model with high accuracy primarily needs an extensive database that is broad-spectrum applicable to various kinds of novel chemicals. The DIPPR 801 database is the most used property database which has a specific hazardous property database. However, these data sets only include the properties of pure chemicals, and the database's hazard and safety table only contains data for FP, AIT, LFL, and UFL. This is also one of the main reasons that most hazardous property QSPR studies are limited to certain properties.

As for regression methods, multiple linear regression (MLR) used to be the standard approach for constructing the QSPR model which shows satisfactory predictability and reliability.[58−60] However, the relationship between descriptors and designated properties is recognized as nonlinear, and the interaction mechanism remains unknown. Forcing a linear regression to the data set will oversimplify the relationship and lead to inaccurate results.[61] Due to this major limitation, there is a significant amount of work that aims to improve the MLR model by implementing particle swarm optimization (PSO) and genetic function approximation (GFA).[62,63] There is an increasing trend of implementing the machine learning algorithm into hazardous property QSPR model construction, which shows significantly superior performance to linear-based models. However, the algorithms used in the QSPR study are still limited to SVM and ANN. This is mainly because the MATLAB software has a machine learning toolbox which has the LIBSVM and neural network tool preinstalled. The graphical user interface is also easier to use compared to programming languages such as Python and R. Jiao et al. and Yuan et al. both compared different machine learning algorithms in QSPR model development, and the result

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

Table 2. Summary of Machine Learning-Based QSAR Studies

name	year	validation	method	activity	software	package/module	ref.
Ai et al.	2019	*k*-fold	RF/SVM/GB	acute toxicity/LC50	R	randomForest/kernlab/Xgboost	97
Bahler and Stone	2000	*k*-fold	NB/ANN	carcinogenicity	Aspirin/MIGRAINES		98
Banerjee et al.	2018	*k*-fold	SVM/RF	acute toxicity/LD50	Python	Scikit-learn	89
Basak et al.	2000	LOO	ANN	acute toxicity/LC50	SAS	VARCLUS	99
Braga et al.	2017	*k*-fold	RF	skin sensitization	Python	Scikit-learn	90
Cui et al.	2015	*k*-fold	ANN/SVM/*k*-NN	respiratory sensitization		OCHEM	88
Dik et al.	2015	LOOCV	SVM/RF	respiratory sensitization	R		91
Fan et al.	2018	*k*-fold	NB/ANN/RF/SVM	acute toxicity/LD50	Orange		100
Forreryd et al.	2015	LOOCV	SVM	respiratory sensitization	R	e1071	92
Guan et al.	2018	*k*-fold	RF/GB/ANN	carcinogenicity	WEKA		101
Lei et al.	2017	*k*-fold	SVM/RF/GB/NB/LDA	repiratory toxicity	R	Xgboost/caret	102
Li et al.	2014	*k*-fold	*k*-NN/SVM/RF	acute toxicity/LD50	Orange		103
Li et al.	2015	validation set	SVM/RF/*k*-NN/NB	carcinogenicity	Orange	LIBSVM	104
Liu et al.	2018	validation set	RF/DNN	acute toxicity/LD50	Python	Keras/Theano	93
Lee et al.	1995	*k*-fold	knowledge-based learning	carcinogenicity		RL4	14
Luechtefeld et al.	2015	validation set	RF	skin sensitization	Python	Scikit-learn	105
Luechtefeld et al.	2018	LOO/*k*-fold	LR/RF/*k*-NN	acute toxicity/LD50	Python	Spark	106
Martin et al.	2017	LOOCV	LDA/HCA	acute toxicity/LD50			107
Moorthy et al.	2017	*k*-fold	RF	carcinogenicity/mutagenicity	WEKA		108
Ren et al.	2006	*k*-fold	LDA/SVM	skin sensitization	R	LIBSVM	109
Sheffield and Judson	2019	validation set	RF/GB/SVM	acute toxicity/LC50	R	ranger/Xgboost/liquidSVM	110
Strickland et al.	2015	LOOCV	LDA/LR/SVM/ANN/NB	skin sensitization	R	nnet/MASS/e1071	111
Strickland et al.	2016	LOOCV	LR/SVM	skin sensitization	R	e1071/randomForest/MASS	112
Tan et al.	2010	hold-out/*k*-fold	SVM/ANN	acute toxicity/LC50			113
Tan et al.	2009	*k*-fold/hold-out	SVM/ANN	carcinogenicity			84
Tung et al.	2019	LOOCV	RF	skin sensitization	R	extraTrees	114
Wang et al.	2010	*k*-fold	SVM	acute toxicity/LC50		LIBSVM	115
Wang et al.	2017	*k*-fold	ANN/SVM/NB/RF/*k*-NN	eye irritation/corrosion	Orange	LIBSVM	96
Wang et al.	2020	*k*-fold	SVM/RF/*k*-NN/CNN	carcinogenicity	Python	TensorFlow/Scikit-learn	94
Wilm et al.	2019	*k*-fold	RF/SVM	skin sensitization	Python	Scikit-learn	116
Xu et al.	2017	validation set	CNN	acute toxicity/LD50	Python	Scikit-learn	117
Yuan et al.	2009	*k*-fold	SVM	skin sensitization	Java	LIVSVM	118
Zang et al.	2016	LOOCV	LDA/SVM/RF	skin sensitization	R	e1071/randomForest/MASS	119
Zhang et al.	2016	*k*-fold	NB	carcinogenicity	Discovery Studio		120
Zhang et al.	2017	*k*-fold	SVM/RF/GB	carcinogenicity	R	caret/randomForest/Xgboost/kernlab	95

showed that the RF- or GB-based model tends to outperform most used SVM methods.[52,64] There is also an advanced neural network algorithm, graphical neural network (GNN), which still has not been implemented into hazardous property QSPR analysis.[65] Besides, it can be also observed from the table that the studies using MATLAB as the regression tool only use the validation set approach for cross-validation. This is due to the lack of the degree of freedom when using proprietary software.

The generic algorithm (GA) is the commonly used descriptor screening method in hazardous property-related QSPR research. Venkatraman et al. used an RF-based VSURF package to conduct descriptor selection.[66] Jiao et al. used the feature importance function of the Xgboost package to rank the descriptor importance based on the contribution in GB tree model development, which is also an interesting way of conducting descriptor screening.[61] Some studies also used the QSPR concept to develop a novel hazardous chemical rating system to improve the hazardous rating and subsequently used it in risk assessment in process industries.[67,68]

For the future development of hazardous property QSPR analysis, more state-of-the-art machine learning algorithms need to be implemented into the descriptor screening process and regression process to further improve the QSPR model's prediction accuracy. The QSPR analysis of hazardous properties also needs to use *k*-fold or leave-one-out cross-validation to improve the credibility of the result. It is worth noting that, in most of the hazardous property QSPR models, although report using the leave-one-out cross-validation method, the validation set approach is used for core statistical value reporting. Besides, due to the difficulty of constructing a database that is large enough for a valid QSPR model development, a more comprehensive database also needs to be constructed based on an exhaustive literature review. This is why only limited QSPR studies focused on mixture properties. More work needs

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

Table 3. Summary of Machine Learning-Based Consequence Analysis

name	year	validation	method	data set	application	software	package	ref.
Cho et al.	2018	validation set	ANN/RF	CFD dispersion simulation	leak source tracking	Python	Keras/Scikit-Learn	134
Jiao et al.	2020	validation set	GB	PHAST simulation	dispersion prediction	R	Xgboost	133
Kim et al.	2018	validation set	ANN	CFD simulation	leak source tracking	Python	Keras/Scikit-Learn	135
Ma et al.	2016	validation set	ANN/SVM	Project Prairie Grass	dispersion prediction and leak source tracking	MATLAB	Neural Network Toolbox/LIBSVM	127
Ni et al.	2019	validation set	DBN/CNN	Project Prairie Grass data set	dispersion prediction	MATLAB	Neural Network Toolbox	128
Qian et al.	2019	validation set	ANN/SVM	Project Prairie Grass data set	dispersion prediction	Python/MATLAB		129
Qiu et al.	2018	validation set	ANN	Indianapolis field data set	dispersion prediction	MATLAB	Neural Network Toolbox	136
Shi et al.	2019	validation set	ANN	FLACS simulation	dispersion prediction			130
Shi et al.	2020	validation set	ANN	FLACS simulation	explosion risk analysis			131
Sun et al.	2019	validation set	ANN	PHAST simulation	fire consequence prediction	MATLAB	Neural Network Toolbox	132
Wang et al.	2015	validation set	ANN	PHAST simulation	dispersion prediction	MATLAB	Neural Network Toolbox	125
Wang et al.	2018	validation set	ANN/SVM	Project Prairie Grass and Indianapolis field data set	dispersion prediction	MATLAB	Neural Network Toolbox/LIBSVM	126

to focus on finding the optimal mixing rule for descriptor calculations.

The QSAR analysis method was developed before the QSPR method. It has been more extensively used in drug discovery which aims to find the chemical with good inhibitory effects on specific targets and low toxicity.[80] For QSAR studies in chemical health and safety areas, the activity is mostly referred to as the toxicity of chemicals.[81]

The QSAR model for toxicity can be categorized into two major groups—acute toxicity prediction and chronic toxicity prediction. The acute toxicity mainly refers to LD_{50} (lethal dose 50) and LC_{50} (lethal concentration 50). The LD_{50} is defined as the amount of chemical that causes 50% (one-half) death in a group of test animals when ingested, injected, or applied to the skin of a test animal. LC_{50} is the concentration of the chemical in the air or water that kills 50% of the test animals with a single exposure. Chronic toxicity includes toxicity to reproduction, mutagenicity, and carcinogenicity.[82−85] Other than toxicology, there are also QSAR studies about skin sensitization and respiratory sensitization which are defined as the allergic response to a substance after skin contact/inhalation.[86−88] Compared to irritation, sensitization is an immunological response to exposure to chemicals. It also worth noting that the QSAR analysis on acute toxicity is mostly regression analysis based on the dependent variable, LC_{50} or LD_{50}. Other chronic toxicities and sensitization analyses are mostly classification analyses.

The summary of machine learning-based QSAR studies in the chemical health and safety area is shown in Table 2. Compared to the QSPR study on hazardous properties, there are many more studies that focus on the QSAR study on toxicology. There are several reasons for the imbalance of research focuses on each area. First, the toxicology experimental study requires animal experiments, which is much more complicated and time-consuming than chemical property measurement. Therefore, there is a much stronger demand for an experiment-free toxicology prediction tool and novel drug development and chemical hazard assessment. In addition, the QSPR analysis is a derivative of QSAR, and QSAR for toxicology is also a more well-studied interdisciplinary area with combining focus on health science and biology. Besides, due to the intensive research on toxicology, there are many more publicly available databases for model development. The QSAR/QSPR analysis is fundamentally a data-intensive method, which requires largely available experimental data to construct a valid database. For the toxicology study, there are several comprehensive databases such as TOXNET, Tox21, and DrugBank which are organized databases readily available for model development.[82] For the reasons mentioned above, the machine learning-based toxicology QSAR analysis is more well-developed compared to the other machine learning-based research in the chemical health and safety area.

From Table 2, we can see that most of the toxicology research implemented the *k*-fold cross-validation method or leave-one-out cross-validation, which is a huge step forward compared to hazardous property QSPR study. For descriptor selection, the QSAR studies also implement different feature selection methods such as random forest-based descriptor importance. Besides, the analysis tool used for QSAR analysis is not limited to MATLAB. More open-source software such as Python and R is used to provide a more flexible operating environment for model development.[89−92] Certain research also used more advanced deep learning tools such as Keras and TensorFlow in model development which also shows superior performance to relatively conventional machine learning methods.[93,94] The gradient boosting-based Xgboost package is more frequently used, which also proved to be the best-performing machine learning algorithm for QSAR/QSPR study.

Furthermore, in order to overcome the disadvantage of machine learning-based regression models not having an equation for application, some studies take a step further by developing a web-based graphical user interface (GUI). The GUI is available for an intuitive application, which can make the application of the machine learning model not limited to simple model training and development and can also help the

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

Table 4. Summary of Neural Network-Based FDD Studies

author	year	methods	features	advantages	systems applied	ref.
Chen et al.	2015	CNN	CNN	outstanding performance compare with SVM	gearbox	146
Kim et al.	2011	FFNN	multiplayer FFNN with backpropagation of errors	enhance sensitivity to leak condition	plasma processing equipment	143
Lee et al.	2017	CNN	CNN	locate the variable and time information that represents process faults	chemical vapor deposition process	147
Othman et al.	2007	HNN	two stages of the HNN	shorten the training time and narrow the diagnostic focus	fatty acid fractionation precut column	138
Pan et al.	2020	RNN	LSTM-based RNN	demonstrate the capability of LSTM model in the fault detection of a chemical looping system	chemical looping systems	149
Pratama et al.	2013	FNN	dynamic fuzzy neural network	high predictive accuracy and low structural complexity	ball nose end milling process	142
Rostek et al.	2015	FFNN	multilayer FFNN	early detection and prediction of leaks and coarse fault isolation	fluidized-bed boilers	144
Ruiz et al.	2001	FNN	FNN	accelerate the diagnosis and improve the performance of the PFD&D system	fluidized bed coal gasifier	140
Rusinov et al.	2009	HNN	HNN with PCA	shorten the training time	hydrocarbon pyrolysis process	139
Shahnazar	2020	RNN	RNN	do not require the existence of plant fault history or first principles models	CSTR	148
Wang et al.	2007	FNN	FNN combined with genetic algorithm (GA)	do not require complicated knowledge of the reaction mechanisms, kinetics, mass, and heat transfer	fluidized-bed catalytically cracked gasoline	141

researcher gather more data for further improvement of the models.[89,95] For the future direction, more research should focus on catching the trend of state-of-the-art machine learning and deep learning algorithms by using more toolkits such as TensorFlow and Pytorch. More research needs to focus on the mixture toxicology analysis. The machine learning-based QSAR analysis also needs to expand toward different chemical health and safety-related activities such as eye irritation.[9]

4.2. Consequence Analysis. For consequence analysis, the current mainstream method is computational fluid dynamics (CFD) modeling.[121−123] With the development of ML algorithms, the artificial neural network (ANN) has been widely used in consequence prediction like gas dispersion and source terms estimation. It could also be integrated with other dispersion models such as PHAST or FLACS to overcome the limitation of missing source information in emergency cases.[124] The summary of ML-based consequence analysis studies is shown in Table 3.

Wang et al. developed the backpropagation neural network to correlate the detected gas concentration from gas detectors with the gas concentration in target locations.[125] Therefore, hazardous gas dispersions could be estimated even when the source information is unknown. Among the neural network training processes, the unstable selection of input parameters may increase the difficulty in model training and consequently reduces the prediction accuracy. To solve this problem, Wang et al. used the integrated Gaussian parameters rather than the original monitoring parameters for both backpropagation neural networks (BPNNs) and SVM models, which improved prediction accuracy.[126] Ma et al. tested a series of machine learning algorithm models combined with a classic Gaussian model. It turned out that the Gaussian-SVM model performs best, and its computation time was close to that of the classic Gaussian dispersion model.[127]

Since 2010, deep learning has gained much popularity due to its supremacy in terms of accuracy when trained with a huge amount of data from the dispersion model. Ni et al. introduced the deep belief networks (DBNs) and convolution neural networks (CNNs) to solve the conflict between accuracy and efficiency requirements of the gas dispersion model.[128] Qian et al. proposed a specially designed LSTM model directly to gas dispersion in the real environment. The dropout technique was used to prevent overfitting and to improve the generalization ability of the model.[129] Shi et al. implemented ANN to both explosion risk analysis and flammable dispersion by using FLACS simulation results as the data set, which can also provide a reliable result to further improve the field of study.[130,131] There are also studies trying to implement the concept of QSAR/QSPR into consequence analysis modeling by using parameters in consequence modeling as property descriptors and making the consequence values as target variables. Sun et al. and Jiao et al. used PHAST simulation to construct a consequence database for fire radiation distance and flammable dispersion and used them to train the model to develop a quantitative property−consequence relationship (QPCR) model which can efficiently predict the corresponding consequence results.[132,133]

The biggest limitation of ML-based consequence analysis is the lack of experimental data since it is extremely expensive and hazardous to conduct field dispersion experiments. The current studies try to tackle this problem by using widely validation dispersion simulation tools like CFD models of COMSOL, FLACS, and ANSYS or an integrated model like PHAST, and hence, the simulation will generate data for model training and validation. However, the simulation tools cannot take account of complex environmental parameters that can highly influence the consequence. Therefore, more experimental data is still needed for further model development.

4.3. Fault Detection and Diagnostics. *4.3.1. Methodology Study.* Neural network-based methods treat fault detection and diagnostics (FDD) as classification problems. For fault detection, networks divide process data into two classes, normal or faulty. For fault diagnostics, fault data is further divided into several classes. The training data is pairs of symptoms and corresponding faults. The trained weight is stored after the neural network determines the relationship between a specific symptom and its corresponding faults.[137] The trained network can then be used to diagnose faults in the test set.

The neural network provides an effective way to obtain information from a large amount of empirical data without the

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

Table 5. Summary of Neural Network-Based TE Process Studies

author	year	methods	features	advantages	ref.
Chadha et al.	2017	DNN	DSN/SAE	comparison between deep stacking networks and sparse stacked autoencoders, SAE architecture has superior performance on all the fault cases on average	160
Chadha et al.	2019	CNN	DCNN	increase the sample efficiency of the training procedure	168
Deng et al.	2017	DNN	DL mechanism with the PCA and KPCA components	superior performance over the existing state-of-the-art KPCA approach	155
Eslamloueyan	2011	ANN	DOHANN	diagnose the faults considerably better than single neural network and DPCA	151
Jin et al.	2018	DNN	WA-DNN	able to learn the high-level and complex features of the sensors' signals	162
Karimi et al.	2014	ANN	SAMMELF	ensemble of classifiers to reduce or remove the risk of choosing an inappropriate single classifier	152
Kumari et al.	2020	BM	HBM	handle source-to-source variability and minimize uncertainty in root cause analysis of rare events	169
Rad et al.	2015	ANN	SLMLP classifier with ICA	improve the performance of single neural network by dividing the fault pattern space into a few smaller subspaces	153
Ren et al.	2019	CNN	DRCNN	give complete as well as in-depth feature representation, and it is easily optimized through the blocks structure	166
Wang et al.	2018	RNN	LSTM-based RNN together with the Euclidean detector	detect various types of data injection attacks	158
Wei et al.	2020	DBN	DBN with drop out technology	reduce overfitting and improve the generalization ability of the model	164
Wu et al.	2018	CNN	DCNN	extract local patterns or features better, has fewer parameters and requires less computation time	165
Wu et al.	2020	TL	TL	overcome the fault data rareness and no label issues in some modes	167
Xie et al.	2015	DNN	HDNN	combine DNN and HNN, better than SNN and DOHANN	154
Zhang et al.	2017	DBN	DBN with OCON architecture	excellent performance due to its feature extraction capability	163
Zhang et al.	2019	CNN	DCNN	detect microfaults efficiently and distinguish different faults from normal cases precisely	170
Zhang et al.	2020	RNN	BiRNN	facilitate the perception of variable deviations over all time points, particularly when a fault has just happened	159
Zhao et al.	2018	RNN	LSTM	learn the representation of raw input data and classifier simultaneously; solve the covariate shift problem	157
Zheng et al.	2020	DNN	SAE	combine feature extraction, data visualization and clustering techniques	161

need for an explicit model. Each type of neural network has advantages and disadvantages. More importantly, they are related to the characteristics of the system being modeled. This section provides a review of the applications of neural network-based FDD methods in chemical process systems, including the artificial neural network (ANN), hierarchical neural network (HNN), fuzzy neural network (FNN), feedforward neural network (FFNN), convolutional neural network (CNN), recurrent neural network (RNN), and their combinations, as summarized in Table 4.

The hierarchical neural network (HNN) and fuzzy neural network (FNN) are two approaches to improve the performance of a single hidden layer neural network. Othman et al. and Rusinov et al. used the HNN to operate faults diagnosis.[138,139] This strategy was able to shorten the training time and narrow the diagnostic focus of the system from node to type of fault. Even if the cause of the abnormal situation was not recognized by the lower-level neural network, the location information was obtained by the higher-level neural network. Ruiz et al. and Wang et al. used the FNN to investigate the faults of fluidized bed coal gasifier and fluidized bed catalytically cracked gasoline.[140,141] FNN can avoid the subjectivity to the definitions of membership function and overcome the gray-box limitation of the neural network by fuzzy logic. The state of the art of the fuzzy neural network was developed by Pratama et al. to achieve a balance between the predictive accuracy and the structural cost.[142]

Due to the emergence of deep learning in 2006, significant progress has been made in the feedforward neural network (FFNN), convolutional neural network (CNN), and recurrent neural network (RNN).[143,144] The deep convolutional neural network (DCNN) has drawn great attention since 2012.[145] It contains convolutional layers and pooling layers for feature extraction. With dropout and pooling layers, the overfitting problem can be greatly avoided. Chen et al. conducted a fault pattern diagnosis of gearboxes using the statistical measures collected from time and frequency domains as the CNN input data.[146] Lee et al. operated a CNN-based fault classification and fault diagnosis method on multivariate sensor signals for semiconductor process monitoring.[147] It extracted a corelationship among a plurality of variables rather than multivariate time-series data.

RNN is FFNN with a time twist. Neurons are trained not just from the previous layer but also from themselves. Long short-term memory networks (LSTMs) are a special kind of RNN designed to avoid the long-term dependency problem. Shahnazar proposed a fault detection and isolation (FDI) methodology that enables the diagnose of single, multiple, and simultaneous actuators and sensor faults without the existence of a first-principles model or plant fault history.[148] Pan et al. presented the LSTM-based RNN to detect the tendency of arching in the standpipe of a chemical looping system.[149]

Although neural network-based FDD methods are really powerful in chemical process systems, some limitations still exist. The first limitation is the lack of data. The performance of the neural network strongly relies on the amount and distribution of the labeled fault data. However, collecting and understanding complex industrial data can be extremely expensive and time-consuming. A potential method to solve this problem is to use simulated data with a transfer learning algorithm. The simulated data with acceptable reliability can be generated based on first-principles knowledge. The transfer

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

learning algorithm can avoid the mismatch between simulated data and actual plant measurements. The second limitation is the difficulty to detect new types of faults of the system and to diagnose faults that happened at the same time. Future studies should focus on the development of monitoring methods that can handle high-dimensional and noisy data by extracting the essential independent components.

4.3.2. Case Study. The Tennessee Eastman (TE) process was created by the Eastman Chemical Company to provide a realistic industrial process for evaluating process control and monitoring.[150] The process consists of five major units: reactor, condenser, compressor, separator, and stripper. The process contains 41 measurements (22 process measurements and 19 composition measurements) and 12 manipulated variables. All measurements have Gaussian noise. The components, kinetics, and operating conditions can be modified. The TE process is an important nonlinear open-loop unstable process that has been used in many studies since it was modeled based on a real process. It is a suitable example to show the development of the neural network, as summarized in Table 5.

For the TE process, there are two main stages of the research, which depends on whether the deep neural network is applied or not. The first stage involves the Bayesian model (BM), single artificial neural network (ANN), hierarchical neural network (HNN), duty-oriented hierarchical artificial neural network (DOHANN), self-adaptive growing neural network (SAGNN), stage-wise additive modeling using a multiclass exponential loss function (SAMMELF), and supervised local multilayer perceptron (SLMLP). The second stage involves a hierarchical deep neural network (HDNN), autoencoder (AE), deep stacking networks (DSN), stacked autoencoders (SAEs), deep convolution neural network (DCNN), deep belief network (DBN), and recurrent neural network (RNN). Recent studies also analyzed the transfer learning (TL) that adapted the difference of data distribution between the different domains.

In stage 1, Eslamloueyan proposed DOHANN, in which each neural network in the HNN has been given a specific duty.[151] The simulation results indicated that DOHANN diagnoses TE process faults much better than single neural network and dynamic principal component analysis (DPCA) methods. To further enhance the diagnosis performance, Karimi et al. proposed an ensemble model that combined several simple (weak) classifiers that relied on the SAMMELF algorithm.[152] The model could reduce or remove the risk of choosing an inappropriate single classifier. Rad et al. developed an SLMLP classifier integrated with two independent component analysis (ICA) models that could detect new types of faults of the system.[153]

In stage 2, Xie et al. integrated DNN with HNN. Compared with the traditional HNN, the HDNN provided a better representation of features from data since the network architecture allowed latent variable space, which resulted in good performance of fault diagnosis.[154] The experiment result showed that the performance of HDNN was much better than the single neural network and DOHANN. Deng et al. proposed a deep principal component analysis (DePCA) method which improved Chen et al.'s method.[155,156] The fault detection rates, fault detection times, and false alarming rates were superior to the existing state-of-the-art PCA and KPCA approaches.

Since the conventional methods suffered from the expertise of feature extraction and classifier design, the LSTM-based RNN is developed by Zhao et al. and Wang et al. Due to the usage of LSTM, the dynamic information on process data could be adaptively utilized.[157,158] Zhang et al. added an additional backward RNN for more feature extraction.[159] The BiRNN facilitated the perception of variable deviations over all time points, especially when a fault just happened so that the fault detection rates increased, and fault detection times decreased. In order to learn the robust frequency-domain features that could not be achieved by conventional networks, the AE-based DNN was developed by Chadha et al. and Zheng et al. SAE architecture exhibited a superior performance on all the fault cases on average over DSN.[160,161] Also, the WA-DNN algorithm introduced by Jin et al. showed the ability to take the time continuity in consideration by analyzing the serial signal with wavelons that SAE was unable to.[162]

In the latest study, an extensible DBN-based fault diagnosis model was proposed by Zhang et al.; the average fault diagnosis rate for the 20 fault types in the TE process reached above 80%.[163] Combining with the dropout technology would further enhance the fault diagnosis rate by learning the global features of the input data. Dropout was used to randomly deactivate each neuron with probability p before the input of each hidden layer so that the generalization ability of the model was improved.[164]

DCNN appeared to be another promising approach with the advantages of better feature extraction and less computation time compared with the DBN. Wu et al. proposed the DCNN model with the average fault detection rates of all the 20 fault types reaching about 90%.[165] To avoid degradation caused by too many layers in the DCNN, Ren et al. introduced a DRCNN model that improves the DCNN model by building blocks that were composed of convolutional layers and identity mappings.[166] It showed a significantly low average false alarm rate.

After all of these powerful deep neural networks were applied to the system, there was a serious problem. Due to the large amount of data with different distribution needed for training, data effectively used in Mode A may not be effectively applied to Mode B. Wu et al. presented a TL-based method for the target mode with the assistance of the source mode to overcome the fault data rareness and no label issues.[167] The experimental results also proved that CNN had better transferability and generality from one mode to another mode than LSTM in the TE process.

5. CHALLENGE AND FUTURE DIRECTION

Although significant progress has been made in the past decade for ML/DL-based chemical health and safety research, ML/DL-based studies in this field are still lagging on implementation of state-of-the-art algorithms, and the research topics are still limited to mostly regression modeling. ML algorithms, which are comparatively simple with easy-to-use toolboxes like MATLAB, are more prevalently used in chemical health and safety research. DL-based algorithms have shown its superior performance in recent years, especially after the toolbox development progression such as PyTorch and TensorFlow. However, the development of a complex DL model still requires a high level of programming skill as well as fundamental algorithm knowledge. Besides, some of the research still lacks statistical validity due to the insufficient

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

experience of statistical modeling, which also needs to be improved in the future.

Another challenge posed for ML/DL-based research in chemical health and safety research is the lack of data for reliable model development. Due to the fact that the experiments in this area are more dangerous and expensive, most of the research only focuses on limited available data sets and areas. One way to tackle this problem is to use reliable and widely validated simulation tools to generate large data sets. However, the accuracy of the model still needs to be further validated against experimental data for a confident result.

Besides, the deployment of the developed ML/DL models is also very important. Only a deployed model with an intuitive and easy-to-use interface can be truly beneficial to the practical implementation, which is particularly challenging since it would require the researchers to be more knowledgeable in programming.

In the field of chemical health and safety, there are still many areas where ML/DL can be applied. For example, natural language processing of archived data and reports can assist risk assessment. Image recognition can identify potential safety hazards and operation abnormality. The areas where the ML/DL models have been widely implemented still need further improvement of algorithms and databases. More state-of-the-art ML/DL toolboxes and algorithms need to be applied and compared to find the suitable one for each subarea of study. More importantly, we need to make sure that the ML/DL models are correctly and appropriately applied without causing problems such as overfitting and bias. This requires researchers and professionals to have a more solid knowledge of the ML/DL principles and algorithms.

6. CONCLUSION

In this Review, important concepts of machine learning, deep learning, and the areas where ML and DL algorithms are currently implemented are comprehensively summarized and analyzed. This Review can provide guidance for scientists, engineers, EHS professionals, and anyone who is involved in chemical health and safety and wants to learn and use ML/DL in their work and research. ML/DL is gradually changing our way of thinking and working, and they will continue to be the hotspot for future research and development. Chemical health and safety, as important subjects to ensure and promote the safety and health of personnel in both academia and industry, still has a long way to go in keeping up with the rapid development of computer science and AI.

AUTHOR INFORMATION

Corresponding Author

Qingsheng Wang − *Mary Kay O'Connor Process Safety Center, Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, Texas 77843-3122, United States;* orcid.org/0000-0002-6411-984X; Email: qwang@tamu.edu

Authors

Zeren Jiao − *Mary Kay O'Connor Process Safety Center, Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, Texas 77843-3122, United States;* orcid.org/0000-0002-2707-0346

Pingfan Hu − *Mary Kay O'Connor Process Safety Center, Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, Texas 77843-3122, United States*

Hongfei Xu − *Mary Kay O'Connor Process Safety Center, Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, Texas 77843-3122, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.chas.0c00075

Author Contributions

†Z.J. and P.H. contributed equally to this work.

Notes

The authors declare no competing financial interest.

ABBREVIATIONS

ABC	artificial bee colony
AE	autoencoder
AI	artificial intelligence
AIT	autoignition temperature
ANN	artificial neural network
API	application programming interface
BRANN	Bayesian regularization artificial neural network
BiRNN	bidirectional recurrent neural network
BM	Bayesian models
BPNN	backpropagation neural network
CFD	computational fluid dynamics
CNN	convolutional neural network
CPU	central processing unit
CRAN	comprehensive r archive network
CSTR	continuous stirred-tank reactor
CUSUM	cumulative sum
DBN	deep belief network
DBSCAN	density-based spatial clustering of applications with noise
DCNN	deep convolution neural network
DFNN	deep feedforward neural network
DL	deep learning
DNN	deep neural network
DOHANN	duty-oriented hierarchical artificial neural network
DPCA	dynamic principal component analysis
DRCNN	deep residual convolutional neural network
DRL	deep reinforcement learning
DSN	deep stacking networks
DWT	discrete wavelet transform
FDD	fault detection and diagnostics
FDI	fault detection and isolation
FDS	fault diagnosis system
FFNN	feed-forward neural network
FP	flash point
FNN	fuzzy neural network
GA	generic algorithm
GB	gradient boosting
GC	gas chromatography
GFA	genetic function approximation
GNN	graphical neural network
GUI	graphical user interface
HBM	hierarchical Bayesian model
HCA	hierarchical clustering analysis
HDNN	hierarchical deep neural network
HNN	hierarchical neural network
ICA	independent component analysis
KBES	knowledge based expert system
k-NN	*k*-nearest neighbor
LC_{50}	lethal concentration 50

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

LD_{50}	lethal dose 50
LDA	linear discrimination analysis
LFL	lower flammability limit
LOOCV	leave-one-out cross-validation
LSTM	long/short-term memory
MIE	minimum ignition energy
ML	machine learning
MLR	multiple linear regression
MPCA	multiway principal component analysis
NB	naïve Bayes
NLP	natural language processing
NNPCA	neural network with principal component analysis
OES	optical emission spectroscopy
PCA	principal component analysis
PSO	particle swarm optimization
QSAR	quantitative structure−activity relationship
QSPR	quantitative structure−property relationship
ReLU	rectified linear unit
RF	random forest
RNN	recurrent neural network
SADT	self-accelerating decomposition temperature
SAE	stacked autoencoders
SAGNN	self-adaptive growing neural network
SAMMELF	stage-wise additive modeling using a multiclass exponential loss function
SANN	Statistica artificial neural network
SLMLP	supervised local multilayer perceptron
SMO	sequential minimal optimization
SVC	support vector clustering
SVM	support vector machine
TEP	Tennessee Eastman process
TL	transfer learning
UFL	upper flammability limit
WA-DNN	wavelons-constructed autoencoder-based deep neural network

■ REFERENCES

(1) Andrieu, C.; Freitas, N.; Doucet, A.; Jordan, M. I. An Introduction to MCMC for Machine Learning. *Mach. Learn.* **2003**, *50*, 5−43.

(2) Samuel, A. L. Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev.* **1959**, *3* (3), 210−229.

(3) Mitchell, T. M. Machine Learning and Data Mining. *Commun. ACM* **1999**, *42* (11), 30−36.

(4) Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Comput. Intell. Neurosci.* **2018**, *2018*, 1−13.

(5) Cambria, E.; White, B. Jumping NLP Curves: A Review of Natural Language Processing Research. *IEEE Comput. Intell. Mag.* **2014**, *9* (2), 48−57.

(6) Chaki, J.; Dey, N.; Shi, F.; Sherratt, R. S. Pattern Mining Approaches Used in Sensor-Based Biometric Recognition: A Review. *IEEE Sens. J.* **2019**, *19* (10), 3569−3580.

(7) Bakator, M.; Radosav, D. Deep Learning and Medical Diagnosis: A Review of Literature. *Multimodal Technol. Interact.* **2018**, *2* (3), 47.

(8) Modi, K.; Dayma, R. Review on Fraud Detection Methods in Credit Card Transactions. *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017.

(9) Chong, E.; Han, C.; Park, F. C. Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, Data Representations, and Case Studies. *Expert Syst. Appl.* **2017**, *83*, 187−205.

(10) Nassif, A. B.; Shahin, I.; Attili, I.; Azzeh, M.; Shaalan, K. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* **2019**, *7*, 19143−19165.

(11) Robertson, G.; Watson, I. A Review of Real-Time Strategy Game AI. *AI Mag.* **2015**, *35* (4), 75.

(12) Pierson, H. A.; Gashler, M. S. Deep Learning in Robotics: a Review of Recent Research. *Adv. Robot.* **2017**, *31* (16), 821−835.

(13) Lecun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521* (7553), 436−444.

(14) Lee, Y.; Buchanan, B. G.; Mattison, D. M.; Klopman, G.; Rosenkranz, H. S. Learning Rules to Predict Rodent Carcinogenicity of Non-Genotoxic Chemicals. *Mutat. Res., Fundam. Mol. Mech. Mutagen.* **1995**, *328*, 127−149.

(15) Pan, Y.; Jiang, J.; Wang, R.; Cao, H. Advantages of Support Vector Machine in QSPR Studies for Predicting Auto-Ignition Temperatures of Organic Compounds. *Chemom. Intell. Lab. Syst.* **2008**, *92* (2), 169−178.

(16) Pan, Y.; Jiang, J.; Wang, R.; Cao, H.; Cui, Y. A Novel QSPR Model for Prediction of Lower Flammability Limits of Organic Compounds Based on Support Vector Machine. *J. Hazard. Mater.* **2009**, *168* (2−3), 962−969.

(17) Witten, I. H.; Frank, E.; Hall, M. A.; Pal, C. J. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Amsterdam, 2017.

(18) James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*; Springer: New York, 2017.

(19) Lewis, D. D. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. *Machine Learning: ECML-98 Lecture Notes in Computer Science* **1998**, *1398*, 4−15.

(20) Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems. *Ann. Eugen.* **1936**, *7*, 179−188.

(21) Cover, T.; Hart, P. E. Nearest Neighbor Classification. *IEEE Trans. Inf. Theory* **1967**, *13* (1), 21−27.

(22) Drucker, H.; Burges, C. J.; Kaufman, L.; Smola, A. J.; Vapnik, V. Support Vector Regression Machines. *Adv. Neural Inf. Process Syst. (NIPS 1996)* **1997**, *9*, 155−161.

(23) Ben-Hur, A.; Horn, D.; Siegelmann, H. T.; Vapnik, V. Support Vector Clustering. *J. Mach. Learn. Res.* **2001**, *2* (Dec), 125−137.

(24) Chang, C.-C.; Lin, C.-J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2* (3), 1−27.

(25) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnnay, E. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(26) Meyer, D.; Dimitriadou, E.; Hornik, K.; Weingessel, A.; Leisch, F.; Chang, C. C.; Lin, C. C.; Meyer, M. D. *Package 'e1071'*; CRAN, 2019.

(27) Ho, T. K. Random Decision Forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995; pp 278−282.

(28) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45* (1), 5−32.

(29) Scornet, E. Random Forests and Kernel Methods. *IEEE Trans. Inf. Theory* **2016**, *62* (3), 1485−1500.

(30) Chen, T.; Guestrin, C. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

(31) Breiman, L. *Arcing the Edge*; Technical Report 486; Statistics Department, University of California at Berkeley: Berkeley, CA, 1997.

(32) Friedman, J. H. Stochastic Gradient Boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367−378.

(33) Nielsen, D. *Tree boosting with xgboost-why does xgboost win "every" machine learning competition?* Master Thesis, Norwegian University of Science and Technology, 2016.

(34) Hartigan, J. A.; Wong, M. A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1979**, *28* (1), 100−108.

(35) Hamerly, G.; Elkan, C. Alternatives to the k-Means Algorithm That Find Better Clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management, CIKM '02*, 2002; pp 600−607.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

(36) Johnson, S. C. Hierarchical Clustering Schemes. *Psychometrika* **1967**, *32* (3), 241−254.
(37) Ester, M.; Kriegel, H. P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD-96 Proceedings* **1996**, *96* (34), 226−231.
(38) Schubert, E.; Sander, J.; Ester, M.; Kriegel, H. P.; Xu, X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM T. Database Syst.* **2017**, *42* (3), 1−21.
(39) McCulloch, W. S.; Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5*, 115−133.
(40) Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**, *65*, 386−408.
(41) Rumelhart, D.; Hinton, G.; Williams, R. Learning Representations by back-propagating errors. *Nature* **1986**, *323*, 533−536.
(42) Hinton, G. E.; Osindero, S.; Teh, Y. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527−1554.
(43) Krstajic, D.; Buturovic, L. J.; Leahy, D. E.; Thomas, S. Cross-Validation Pitfalls When Selecting and Assessing Regression and Classification Models. *J. Cheminf.* **2014**, *6* (1), 1−15.
(44) Celisse, A. Optimal Cross-Validation in Density Estimation with the L^2-loss. *Ann. Stat.* **2014**, *42*, 1879−1910.
(45) R Core Team. *R: A Language and Environment for Statistical Computing*; R Core Team: Vienna, Austria, 2013.
(46) Liaw, A.; Wiener, M. Classification and Regression by RandomForest. *R News.* **2002**, *2* (3), 18−22.
(47) Chen, T.; He, T. *xgboost: eXtreme Gradient Boosting, R package version 1.0.0.2*; CRAN, 2020.
(48) Meyer, D.; Dimitriadou, E.; Hornik, K.; Weingessel, A.; Leisch, F.; Chang, C. C.; Lin, C. C.; Meyer, M. D. *Package 'e1071', R package version 1.7-3*; CRAN, 2019.
(49) *R documentation and manual: TensorFlow Package for R (online)*, 2020. https://www.rdocumentation.org/packages/tensorflow (accessed 20 July, 2020).
(50) Oliphant, T. E. Python for Scientific Computing. *Comput. Sci. Eng.* **2007**, *9* (3), 10−20.
(51) Jiao, Z.; Escobar-Hernandez, H. U.; Parker, T.; Wang, Q. Review of Recent Developments of Quantitative Structure-Property Relationship Models on Fire and Explosion-Related Properties. *Process Saf. Environ. Prot.* **2019**, *129*, 280−290.
(52) Jiao, Z.; Yuan, S.; Zhang, Z.; Wang, Q. Machine Learning Prediction of Hydrocarbon Mixture Lower Flammability Limits Using Quantitative Structure-Property Relationship Models. *Process Saf. Prog.* **2020**, *39* (2), e12103.
(53) Verma, J.; Khedkar, V.; Coutinho, E. 3D-QSAR in Drug Design - A Review. *Curr. Top. Med. Chem.* **2010**, *10*, 95−115.
(54) Quintero, F. A.; Patel, S. J.; Muñoz, F.; Mannan, M. S. Review of Existing QSAR/QSPR Models Developed for Properties Used in Hazardous Chemicals Classification System. *Ind. Eng. Chem. Res.* **2012**, *51* (49), 16101−16115.
(55) Wang, B.; Xu, K.; Wang, Q. Prediction of Upper Flammability Limits for Fuel Mixtures Using Quantitative Structure-Property Relationship Models. *Chem. Eng. Commun.* **2019**, *206* (2), 247−253.
(56) Wang, B.; Yi, H.; Xu, K.; Wang, Q. Prediction of the Self-Accelerating Decomposition Temperature of Organic Peroxides Using QSPR Models. *J. Therm. Anal. Calorim.* **2017**, *128* (1), 399−406.
(57) Zhou, L.; Wang, B.; Jiang, J.; Pan, Y.; Wang, Q. Quantitative Structure-Property Relationship (QSPR) Study for Predicting Gas-Liquid Critical Temperatures of Organic Compounds. *Thermochim. Acta* **2017**, *655*, 112−116.
(58) Wang, B.; Zhou, L.; Xu, K.; Wang, Q. Fast Prediction of Minimum Ignition Energy From Molecular Structure Using Simple QSPR Model. *J. Loss Prev. Process Ind.* **2017**, *50*, 290−294.
(59) Wang, B.; Park, H.; Xu, K.; Wang, Q. Prediction of Lower Flammability Limits of Blended Gases Based on Quantitative Structure-Property Relationship. *J. Therm. Anal. Calorim.* **2018**, *132* (2), 1125−1130.
(60) Zhou, L.; Wang, B.; Jiang, J.; Pan, Y.; Wang, Q. Predicting the Gas-Liquid Critical Temperature of Binary Mixtures Based on the Quantitative Structure Property Relationship. *Chemom. Intell. Lab. Syst.* **2017**, *167*, 190−195.
(61) Jiao, Z.; Ji, C.; Yuan, S.; Zhang, Z.; Wang, Q. Development of Machine Learning Based Prediction Models for Hazardous Properties of Chemical Mixtures. *J. Loss Prev. Process Ind.* **2020**, *67*, 104226.
(62) Bagheri, M.; Borhani, T. N. G.; Zahedi, G. Estimation of Flash Point and Autoignition Temperature of Organic Sulfur Chemicals. *Energy Convers. Manage.* **2012**, *58*, 185−196.
(63) Khajeh, A.; Modarress, H. QSPR Prediction of Flash Point of Esters by Means of GFA and ANFIS. *J. Hazard. Mater.* **2010**, *179* (1−3), 715−720.
(64) Yuan, S.; Jiao, Z.; Quddus, N.; Kwon, J. S.-I.; Mashuga, C. V. Developing Quantitative Structure-Property Relationship Models To Predict the Upper Flammability Limit Using Machine Learning. *Ind. Eng. Chem. Res.* **2019**, *58* (8), 3531−3537.
(65) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545−2559.
(66) Venkatraman, V.; Alsberg, B. K. Quantitative Structure-Property Relationship Modelling of Thermal Decomposition Temperatures of Ionic Liquids. *J. Mol. Liq.* **2016**, *223*, 60−67.
(67) Zhang, Z.; Yuan, S.; Yu, M.; Mannan, M. S.; Wang, Q. A Hazard Index for Chemical Logistic Warehouses with Modified Flammability Rating by Machine Learning Methods. *J. Chem. Health Saf.* **2020**, *27* (3), 190−197.
(68) Yuan, S.; Zhang, Z.; Sun, Y.; Kwon, J. S.-I.; Mashuga, C. V. Liquid Flammability Ratings Predicted by Machine Learning Considering Aerosolization. *J. Hazard. Mater.* **2020**, *386*, 121640.
(69) Borhani, T. N. G.; Afzali, A.; Bagheri, M. QSPR Estimation of the Auto-Ignition Temperature for Pure Hydrocarbons. *Process Saf. Environ. Prot.* **2016**, *103*, 115−125.
(70) Gharagheizi, F. An Accurate Model for Prediction of Autoignition Temperature of Pure Compounds. *J. Hazard. Mater.* **2011**, *189* (1−2), 211−221.
(71) Jiao, L.; Zhang, X.; Qin, Y.; Wang, X.; Li, H. QSPR Study on the Flash Point of Organic Binary Mixtures by Using Electrotopological State Index. *Chemom. Intell. Lab. Syst.* **2016**, *156*, 211−216.
(72) Jin, Y.; Jiang, J.; Pan, Y.; Ni, L. Prediction of the Auto-Ignition Temperature of Binary Liquid Mixtures Based on the Quantitative Structure-Property Relationship Approach. *J. Therm. Anal. Calorim.* **2020**, *140* (1), 397−409.
(73) Mallakpour, S.; Hatami, M.; Golmohammadi, H. QSPR Prediction of Thermal Decomposition Property of Non-Vinyl Polymers Having α-Amino Acids Moieties. *Polym. Bull.* **2013**, *70* (2), 715−732.
(74) Mallakpour, S.; Hatami, M.; Khooshechin, S.; Golmohammadi, H. Evaluations of Thermal Decomposition Properties for Optically Active Polymers Based on Support Vector Machine. *J. Therm. Anal. Calorim.* **2014**, *116* (2), 989−1000.
(75) Mirshahvalad, H.; Ghasemiasl, R.; Raoufi, N.; Malekzadeh Dirin, M. A Neural Network QSPR Model for Accurate Prediction of Flash Point of Pure Hydrocarbons. *Mol. Inf.* **2019**, *38* (4), 1800094.
(76) Saldana, D. A.; Starck, L.; Mougin, P.; Rousseau, B.; Creton, B. Prediction of Flash Points for Fuel Mixtures Using Machine Learning and a Novel Equation. *Energy Fuels* **2013**, *27* (7), 3811−3820.
(77) Saldana, D. A.; Starck, L.; Mougin, P.; Rousseau, B.; Pidol, L.; Jeuland, N.; Creton, B. Flash Point and Cetane Number Predictions for Fuel Compounds Using Quantitative Structure Property Relationship (QSPR) Methods. *Energy Fuels* **2011**, *25* (9), 3900−3908.
(78) Wang, B.; Zhou, L.; Xu, K.; Wang, Q. Prediction of Minimum Ignition Energy from Molecular Structure Using Quantitative Structure-Property Relationship (QSPR) Models. *Ind. Eng. Chem. Res.* **2017**, *56* (1), 47−51.
(79) Wang, B.; Zhou, L.; Liu, X.; Xu, K.; Wang, Q. Prediction of Superheat Limit Temperatures for Fuel Mixtures Using Quantitative Structure-Property Relationship Model. *J. Loss Prev. Process Ind.* **2020**, *64*, 104087.
(80) Lill, M. A. Multi-Dimensional QSAR in Drug Discovery. *Drug Discovery Today* **2007**, *12*, 1013−1017.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

(81) Cao, J.; Pan, Y.; Jiang, Y.; Qi, R.; Yuan, B.; Jia, Z.; Jiang, J.; Wang, Q. Computer-Aided Nanotoxicology: Risk Assessment of Metal Oxide Nanoparticles via nano-QSAR. *Green Chem.* **2020**, *22*, 3512−3521.
(82) Wu, Y.; Wang, G. Machine Learning Based Toxicity Prediction: From Chemical Structural Description to Transcriptome Analysis. *Int. J. Mol. Sci.* **2018**, *19* (8), 2358.
(83) Hao, Y.; Sun, G.; Fan, T.; Sun, X.; Liu, Y.; Zhang, N.; Zhao, L.; Zhong, R.; Peng, Y. Prediction on the Mutagenicity of Nitroaromatic Compounds Using Quantum Chemistry Descriptors Based QSAR and Machine Learning Derived Classification Methods. *Ecotoxicol. Environ. Saf.* **2019**, *186*, 109822.
(84) Tan, N.; Rao, H.; Li, Z.; Li, X. Prediction of Chemical Carcinogenicity by Machine Learning Approaches. *SAR QSAR Environ. Res.* **2009**, *20*, 27−75.
(85) Luechtefeld, T.; Rowlands, C.; Hartung, T. Big-Data and Machine Learning to Revamp Computational Toxicology and Its Use in Risk Assessment. *Toxicol. Res.* **2018**, *7*, 732−744.
(86) Kleinstreuer, N. C.; Hoffmann, S.; Alépée, N.; Allen, D.; Ashikaga, T.; Casey, W.; Clouet, E.; Cluzel, M.; Desprez, B.; Gellatly, N.; Gobel, C.; Kern, P. S.; Klaric, M.; Kuhnl, J.; Martinozzi-Teissier, S.; Mewes, K.; Miyazawa, M.; Strickland, J.; van Vliet, E.; Zang, Q.; Petersohn, D. Non-Animal Methods to Predict Skin Sensitization (II): an Assessment of Defined Approaches. *Crit. Rev. Toxicol.* **2018**, *48*, 359−374.
(87) Wilm, A.; Kühnl, J.; Kirchmair, J. Computational Approaches for Skin Sensitization Prediction. *Crit. Rev. Toxicol.* **2018**, *48*, 738−760.
(88) Cui, X.; Yang, R.; Li, S.; Liu, J.; Wu, Q.; Li, X. Modeling and Insights into Molecular Basis of Low Molecular Weight Respiratory Sensitizers. *Mol. Diversity* **2020**, 1−13.
(89) Banerjee, P.; Eckert, A. O.; Schrey, A. K.; Preissner, R. ProTox-II: a Webserver for the Prediction of Toxicity of Chemicals. *Nucleic Acids Res.* **2018**, *46* (W1), W257−W263.
(90) Braga, R. C.; Alves, V. M.; Muratov, E. N.; Strickland, J.; Kleinstreuer, N.; Trospsha, A.; Andrade, C. H. Pred-Skin: A Fast and Reliable Web Application to Assess Skin Sensitization Effect of Chemicals. *J. Chem. Inf. Model.* **2017**, *57*, 1013−1017.
(91) Dik, S.; Pennings, J. L.; Loveren, H. V.; Ezendam, J. Development of an in Vitro Test to Identify Respiratory Sensitizers in Bronchial Epithelial Cells Using Gene Expression Profiling. *Toxicol. In Vitro* **2015**, *30*, 274−280.
(92) Forreryd, A.; Johansson, H.; Albrekt, A.-S.; Borrebaeck, C. A. K.; Lindstedt, M. Prediction of Chemical Respiratory Sensitizers Using GARD, a Novel In Vitro Assay Based on a Genomic Biomarker Signature. *PLoS One* **2015**, *10* (3), e0118808.
(93) Liu, R.; Madore, M.; Glover, K. P.; Feasel, M. G.; Wallqvist, A. Assessing Deep and Shallow Learning Methods for Quantitative Prediction of Acute Chemical Toxicity. *Toxicol. Sci.* **2018**, *164*, 512−526.
(94) Wang, Y.-W.; Huang, L.; Jiang, S.-W.; Li, K.; Zou, J.; Yang, S.-Y. CapsCarcino: A Novel Sparse Data Deep Learning Tool for Predicting Carcinogens. *Food Chem. Toxicol.* **2020**, *135*, 110921.
(95) Zhang, L.; Ai, H.; Chen, W.; Yin, Z.; Hu, H.; Zhu, J.; Zhao, J.; Zhao, Q.; Liu, H. CarcinoPred-EL: Novel Models for Predicting the Carcinogenicity of Chemicals Using Molecular Fingerprints and Ensemble Learning Methods. *Sci. Rep.* **2017**, *7* (1), 1−14.
(96) Wang, Q.; Li, X.; Yang, H.; Cai, Y.; Wang, Y.; Wang, Z.; Li, W.; Tang, Y.; Liu, G. In Silico Prediction of Serious Eye Irritation or Corrosion Potential of Chemicals. *RSC Adv.* **2017**, *7*, 6697−6703.
(97) Ai, H.; Wu, X.; Zhang, L.; Qi, M.; Zhao, Y.; Zhao, Q.; Zhao, J.; Liu, H. QSAR Modelling Study of the Bioconcentration Factor and Toxicity of Organic Compounds to Aquatic Organisms Using Machine Learning and Ensemble Methods. *Ecotoxicol. Environ. Saf.* **2019**, *179*, 71−78.
(98) Bahler, D.; Stone, B.; Wellington, C.; Bristol, D. W. Symbolic, Neural, and Bayesian Machine Learning Models for Predicting Carcinogenicity of Chemical Compounds. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (4), 906−914.
(99) Basak, S. C.; Grunwald, G. D.; Gute, B. D.; Balasubramanian, K.; Opitz, D. Use of Statistical and Neutral Net Approaches in Predicting Toxicity of Chemicals. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (4), 885−890.
(100) Fan, T.; Sun, G.; Zhao, L.; Cui, X.; Zhong, R. QSAR and Classification Study on Prediction of Acute Oral Toxicity of N-Nitroso Compounds. *Int. J. Mol. Sci.* **2018**, *19* (10), 3015.
(101) Guan, D.; Fan, K.; Spence, I.; Matthews, S. Combining Machine Learning Models of in Vitro and in Vivo Bioassays Improves Rat Carcinogenicity Prediction. *Regul. Toxicol. Pharmacol.* **2018**, *94*, 8−15.
(102) Lei, T.; Chen, F.; Liu, H.; Sun, H.; Kang, Y.; Li, D.; Li, Y.; Hou, T. ADMET Evaluation in Drug Discovery. Part 17: Development of Quantitative and Qualitative Prediction Models for Chemical-Induced Respiratory Toxicity. *Mol. Pharmaceutics* **2017**, *14*, 2407−2421.
(103) Li, X.; Chen, L.; Cheng, F.; Wu, Z.; Bian, H.; Xu, C.; Li, W.; Liu, G.; Shen, X.; Tang, Y. In Silico Prediction of Chemical Acute Oral Toxicity Using Multi-Classification Methods. *J. Chem. Inf. Model.* **2014**, *54*, 1061−1069.
(104) Li, X.; Du, Z.; Wang, J.; Wu, Z.; Li, W.; Liu, G.; Shen, X.; Tang, Y. In Silico Estimation of Chemical Carcinogenicity with Binary and Ternary Classification Methods. *Mol. Inf.* **2015**, *34*, 228−235.
(105) Luechtefeld, T.; Maertens, A.; Mckim, J. M.; Hartung, T.; Kleensang, A.; Sá-Rocha, V. Probabilistic Hazard Assessment for Skin Sensitization Potency by Dose-Response Modeling Using Feature Elimination Instead of Quantitative Structure-Activity Relationships. *J. Appl. Toxicol.* **2015**, *35*, 1361−1371.
(106) Luechtefeld, T.; Marsh, D.; Rowlands, C.; Hartung, T. Machine Learning of Toxicological Big Data Enables Read-Across Structure Activity Relationships (RASAR) Outperforming Animal Test Reproducibility. *Toxicol. Sci.* **2018**, *165*, 198−212.
(107) Martin, T. M.; Lilavois, C. R.; Barron, M. G. Prediction of Pesticide Acute Toxicity Using Two-Dimensional Chemical Descriptors and Target Species Classification. *SAR QSAR Environ. Res.* **2017**, *28*, 525−539.
(108) Moorthy, N. H. N.; Kumar, S.; Poongavanam, V. Classification of Carcinogenic and Mutagenic Properties Using Machine Learning Method. *Comput. Toxicol.* **2017**, *3*, 33−43.
(109) Ren, Y.; Liu, H.; Xue, C.; Yao, X.; Liu, M.; Fan, B. Classification Study of Skin Sensitizers Based on Support Vector Machine and Linear Discriminant Analysis. *Anal. Chim. Acta* **2006**, *572*, 272−282.
(110) Sheffield, T. Y.; Judson, R. S. Ensemble QSAR Modeling to Predict Multispecies Fish Toxicity Lethal Concentrations and Points of Departure. *Environ. Sci. Technol.* **2019**, *53*, 12793−12802.
(111) Strickland, J.; Zang, Q.; Paris, M.; Lehmann, D. M.; Choksi, N.; Matheson, J.; Jacobs, A.; Casey, W.; Kleinstreuer, N. Integrated Decision Strategies for Skin Sensitization Hazard. *J. Appl. Toxicol.* **2016**, *36*, 1150−1162.
(112) Strickland, J.; Zang, Q.; Paris, M.; Lehmann, D. M.; Allen, D.; Choksi, N.; Matheson, J.; Jacobs, A.; Casey, W.; Kleinstreuer, N. Multivariate Models for Prediction of Human Skin Sensitization Hazard. *J. Appl. Toxicol.* **2017**, *37*, 347−360.
(113) Tan, N.; Li, P.; Rao, H.; Li, Z.; Li, X. Prediction of the Acute Toxicity of Chemical Compounds to the Fathead Minnow by Machine Learning Approaches. *Chemom. Intell. Lab. Syst.* **2010**, *100*, 66−73.
(114) Tung, C.-W.; Lin, Y.-H.; Wang, S.-S. Transfer Learning for Predicting Human Skin Sensitizers. *Arch. Toxicol.* **2019**, *93*, 931−940.
(115) Wang, Y.; Zheng, M.; Xiao, J.; Lu, Y.; Wang, F.; Lu, J.; Luo, X.; Zhu, W.; Jiang, H.; Chen, K. Using Support Vector Regression Coupled with the Genetic Algorithm for Predicting Acute Toxicity to the Fathead Minnow. *SAR QSAR Environ. Res.* **2010**, *21*, 559−570.
(116) Wilm, A.; Stork, C.; Bauer, C.; Schepky, A.; Kühnl, J.; Kirchmair, J. Skin Doctor: Machine Learning Models for Skin Sensitization Prediction That Provide Estimates and Indicators of Prediction Reliability. *Int. J. Mol. Sci.* **2019**, *20*, 4833.

(117) Xu, Y.; Pei, J.; Lai, L. Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *J. Chem. Inf. Model.* **2017**, *57*, 2672−2685.
(118) Yuan, H.; Huang, J.; Cao, C. Prediction of Skin Sensitization with a Particle Swarm Optimized Support Vector Machine. *Int. J. Mol. Sci.* **2009**, *10*, 3237−3254.
(119) Zang, Q.; Paris, M.; Lehmann, D. M.; Bell, S.; Kleinstreuer, N.; Allen, D.; Matheson, J.; Jacobs, A.; Casey, W.; Strickland, J. Prediction of Skin Sensitization Potency Using Machine Learning Approaches. *J. Appl. Toxicol.* **2017**, *37*, 792−805.
(120) Zhang, H.; Cao, Z.-X.; Li, M.; Li, Y.-Z.; Peng, C. Novel Naïve Bayes Classification Models for Predicting the Carcinogenicity of Chemicals. *Food Chem. Toxicol.* **2016**, *97*, 141−149.
(121) Shen, R.; Jiao, Z.; Parker, T.; Sun, Y.; Wang, Q. Recent Application of Computational Fluid Dynamics (CFD) in Process Safety and Loss Prevention: A Review. *J. Loss Prev. Process Ind.* **2020**, *67*, 104252.
(122) Jiao, Z.; Yuan, S.; Ji, C.; Mannan, M. S.; Wang, Q. Optimization of Dilution Ventilation Layout Design in Confined Environments Using Computational Fluid Dynamics (CFD). *J. Loss Prev. Process Ind.* **2019**, *60*, 195−202.
(123) Mi, H.; Liu, Y.; Jiao, Z.; Wang, W.; Wang, Q. A Numerical Study on the Optimization of Ventilation Mode During Emergency of Cable Fire in Utility Tunnel. *Tunn. Undergr. Space Technol.* **2020**, *100*, 103403.
(124) Ma, D.; Gao, J.; Zhang, Z.; Zhao, H.; Wang, Q. Locating the Gas Leakage Source in the Atmosphere Using the Dispersion Wave Method. *J. Loss Prev. Process Ind.* **2020**, *63*, 104031.
(125) Wang, B.; Chen, B.; Zhao, J. The Real-time Estimation of Hazardous Gas Dispersion by the Integration of Gas Detectors, Neural Network and Gas Dispersion Models. *J. Hazard. Mater.* **2015**, *300*, 433−442.
(126) Wang, R.; Chen, B.; Qiu, S.; Zhu, Z.; Wang, Y.; Qiu, X. Comparison of Machine Learning Models for Hazardous Gas Dispersion Prediction in Field Cases. *Int. J. Environ. Res. Public Health* **2018**, *15*, 1450.
(127) Ma, D.; Zhang, Z. Contaminant Dispersion Prediction and Source Estimation with Integrated Gaussian-machine Learning Network Model for Point Source Emission in Atmosphere. *J. Hazard. Mater.* **2016**, *311*, 237−245.
(128) Ni, J.; Yang, H.; Yao, J.; Li, Z.; Qin, P. Toxic Gas Dispersion Prediction for Point Source Emission using Deep Learning Method. *Hum. Ecol. Risk Assess.* **2020**, *26*, 557.
(129) Qian, F.; Chen, L.; Li, J.; Ding, C.; Chen, X.; Wang, J. Direct Prediction of the Toxic Gas Diffusion Rule in a Real Environment Based on LSTM. *Int. J. Environ. Res. Public Health* **2019**, *16*, 2133.
(130) Shi, J.; Li, X.; Khan, F.; Chang, Y.; Zhu, Y.; Chen, G. Artificial Bee Colony Based Bayesian Regularization Artificial Neural Network Approach to Model Transient Flammable Cloud Dispersion in Congested Area. *Process Saf. Environ. Prot.* **2019**, *128*, 121−127.
(131) Shi, J.; Chang, B.; Khan, F.; Chang, Y.; Zhu, Y.; Chen, G.; Zhang, C. Stochastic Explosion Risk Analysis of Hydrogen Production Facilities. *Int. J. Hydrogen Energy* **2020**, *45*, 13535−13550.
(132) Sun, Y.; Wang, J.; Zhu, W.; Yuan, S.; Hong, Y.; Mannan, M. S.; Wilhite, B. Development of Consequent Models for Three Categories of Fire through Artificial Neural Networks. *Ind. Eng. Chem. Res.* **2020**, *59* (1), 464−474.
(133) Jiao, Z.; Sun, Y.; Hong, Y.; Parker, T.; Hu, P.; Wang, Q. Development of Flammable Dispersion Quantitative Property−Consequence Relationship Models Using Extreme Gradient Boosting. *Ind. Eng. Chem. Res.* **2020**, *59* (33), 15109−15118.
(134) Cho, J.; Kim, H.; Gebreselassie, A. L.; Shin, D. Deep Neural Network and Random Forest Classifier for Source Tracking of Chemical Leaks using Fence Monitoring Data. *J. Loss Prev. Process Ind.* **2018**, *56*, 548−558.
(135) Kim, H.; Yoon, E. S.; Shin, D. Deep Neural Networks for Source Tracking of Chemical Leaks and Improved Chemical Process Safety. *Comput.-Aided Chem. Eng.* **2018**, *44*, 2359−2364.
(136) Qiu, S.; Chen, B.; Wang, R.; Zhu, Z.; Wang, Y.; Qiu, X. Atmospheric Dispersion Prediction and Source Estimation of Hazardous Gas using Artificial Neural Network, Particle Swarm Optimization and Expectation Maximization. *Atmos. Environ.* **2018**, *178*, 158−163.
(137) Zhang, J.; Roberts, P. D. On-line Process Fault Diagnosis using Neural Network Techniques. *Trans. Inst. Meas. Control* **1992**, *14*, 179−188.
(138) Othman, M. R.; Ali, M. W.; Kamsah, M. Z. Process Fault Detection using Hierarchical Artificial Neural Network Diagnostic Strategy. *J. Teknologi.* **2007**, *46*, 11−26.
(139) Rusinov, L. A.; Rudakova, I. V.; Remizova, O. A.; Kurkina, V. V. Fault Diagnosis in Chemical Processes with Application of Hierarchical Neural Networks. *Chemom. Intell. Lab. Syst.* **2009**, *97*, 98−103.
(140) Ruiz, D.; Nougues, J. M.; Puigjaner, L. Fault Diagnosis Support System for Complex Chemical Plants. *Comput. Chem. Eng.* **2001**, *25*, 151−160.
(141) Wang, Z.; Tang, B.; Chen, C.; Yuan, J.; Wang, L. Modeling and Optimization for the Secondary Reaction of FCC Gasoline Based on the Fuzzy Neural Network and Genetic Algorithm. *Chem. Eng. Process.* **2007**, *46*, 175−180.
(142) Pratama, M.; Er, M. J.; Li, X.; Oentaryo, R. J.; Lughofer, E.; Arifin, I. Data Driven Modeling Based on Dynamic Parsimonious Fuzzy Neural Network. *Neurocomputing.* **2013**, *110*, 18−28.
(143) Kim, B.; Kwon, S. Wavelet-coupled Backpropagation Neural Network as a Chamber Leak Detector of Plasma Processing Equipment. *Expert. Syst. Appl.* **2011**, *38*, 6275−6280.
(144) Rostek, K.; Morytko, Ł.; Jankowska, A. Early Detection and Prediction of Leaks in Fluidized-bed Boilers using Artificial Neural Networks. *Energy* **2015**, *89*, 914−923.
(145) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst. (NIPS 2012)* **2012**, *25*, 1097−1105.
(146) Chen, Z.; Li, C.; Sanchez, R. Gearbox Fault Identification and Classification with Convolutional Neural Networks. *Shock Vib.* **2015**, *2015*, 1.
(147) Lee, K. B.; Cheon, S.; Kim, C. O. A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes. *IEEE Trans. Semicond. Manuf.* **2017**, *30*, 135−142.
(148) Shahnazari, H. Fault Diagnosis of Nonlinear Systems using Recurrent Neural Networks. *Chem. Eng. Res. Des.* **2020**, *153*, 233−245.
(149) Pan, J.; Pottimurthy, Y.; Wang, D.; Hwang, S.; Patil, S.; Fan, L. Recurrent Neural Network Based Detection of Faults Caused Byparticle Attrition in Chemical Looping Systems. *Powder Technol.* **2020**, *367*, 266−276.
(150) Ricker, N. L. Decentralized Control of the Tennessee Eastman Challenge. *J. Process Control* **1996**, *6* (4), 205−221.
(151) Eslamloueyan, R. Designing a Hierarchical Neural Network Based on Fuzzy Clustering for Fault Diagnosis of the Tennessee-Eastman Process. *Appl. Soft Comput.* **2011**, *11*, 1407−1415.
(152) Karimi, P.; Jazayeri-Rad, H. Comparing the Fault Diagnosis Performances of Single Neural Networks and Two Ensemble Neural Networks Based on the Boosting Methods. *J. Automat. Control* **2014**, *2*, 21−32.
(153) Rad, M. A. A.; Yazdanpanah, M. J. Designing Supervised Local Neural Network Classifiers Based on EM clustering for fault diagnosis of Tennessee Eastman process. *Chemom. Intell. Lab. Syst.* **2015**, *146*, 149−157.
(154) Xie, D.; Li, B. A hierarchical Deep Neural Network for Fault Diagnosis on Tennessee-Eastman Process. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, 2015; pp 745−748.
(155) Deng, X.; Tian, X.; Chen, S.; Harris, C. J. Deep Learning Based Nonlinear Principal Component Analysis for Industrial Process Fault Detection. In *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2017; pp 1237−1243.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334

(156) Chen, J. H.; Liao, C. M. Dynamic Process Fault Monitoring Based on Neural Network and PCA. *J. Process Control* **2002**, *12*, 277−289.
(157) Zhao, H.; Sun, S.; Jin, B. Sequential Fault Diagnosis Based on LSTM Neural Network. *IEEE Access* **2018**, *6*, 12929−12939.
(158) Wang, W.; Xie, Y.; Ren, L.; Zhu, X.; Chang, R.; Yin, Q. Detection of Data Injection Attack in Industrial Control System using Long Short Term Memory Recurrent Neural Network. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, 2018; pp 2710−2715.
(159) Zhang, S.; Bi, K.; Qiu, T. Bidirectional Recurrent Neural Network-based Chemical Process Fault Diagnosis. *Ind. Eng. Chem. Res.* **2020**, *59*, 824−834.
(160) Chadha, G. S.; Schwung, A. Comparison of Deep Neural Network Architectures for Fault Detection in Tennessee Eastman Process. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, 2017; pp 1−8.
(161) Zheng, S.; Zhao, J. A New Unsupervised Data Mining Method Based on the Stacked Autoencoder for Chemical Process Fault Diagnosis. *Comput. Chem. Eng.* **2020**, *135*, 106755.
(162) Jin, M.; Yang, W.; Wang, Y.; Zhang, H. Wavelons-constructed Autoencoder-based Deep Neural Network for Fault Detection in Chemical Processes. In *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, 2018; pp 332−337.
(163) Zhang, Z.; Zhao, J. A Deep Belief Network Based Fault Diagnosis Model for Complex Chemical Processes. *Comput. Chem. Eng.* **2017**, *107*, 395−407.
(164) Wei, Y.; Weng, Z. Research on TE Process Fault Diagnosis Method Based on DBN and Dropout. *Can. J. Chem. Eng.* **2020**, *98*, 1293−1306.
(165) Wu, H.; Zhao, J. Deep Convolutional Neural Network Model Based Chemical Process Fault Diagnosis. *Comput. Chem. Eng.* **2018**, *115*, 185−197.
(166) Ren, X.; Zou, Y.; Zhang, Z. Fault Detection and Classification with Feature Representation Based on Deep Residual Convolutional Neural Network. *J. Chemom.* **2019**, *33*, e3170.
(167) Wu, H.; Zhao, J. Fault Detection and Diagnosis Based on Transfer Learning for Multimode Chemical Processes. *Comput. Chem. Eng.* **2020**, *135*, 106731.
(168) Chadha, G. S.; Krishnamoorthy, M.; Schuwung, A. Time Series Based Fault Detection in Industrial Processes using Convolutional Neural Networks. In *IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal, 2019; pp 173−178.
(169) Kumari, P.; Lee, D.; Wang, Q.; Karim, M. N.; Kwon, J. S. I. Root Cause Analysis of Key Process Variable Deviation for Rare Events in the Chemical Process Industry. *Ind. Eng. Chem. Res.* **2020**, *59* (23), 10987−10999.
(170) Zhang, H.; Wang, P.; Gao, X.; Qi, Y.; Gao, H. Amplitude-frequency Imagesbased ConvNet: Applications of Fault Detection and Diagnosis in Chemical Processes. *J. Chemom.* **2019**, *33*, e3168.

https://dx.doi.org/10.1021/acs.chas.0c00075
ACS Chem. Health Saf. 2020, 27, 316−334