An Architecture of Deep Learning in QSPR Modeling for the Prediction of Critical Properties Using Molecular Signatures

Yang Su[a], Zihao Wang[a], Saimeng Jin[a], Weifeng Shen[a,*], Jingzheng Ren[b] and Mario R. Eden[c]

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

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

[c]*Department of Chemical Engineering, Auburn University, Auburn, Alabama 36849, United States*

[*]**Corresponding author:** (W. S.) shenweifeng@cqu.edu.cn

Abstract: Deep learning rapidly promotes many fields with successful stories in natural language processing. An architecture of deep neural network (DNN) combining tree-structured long short-term memory (Tree-LSTM) network and back-propagation neural network (BPNN) is developed for predicting physical properties. Inspired by the natural language processing in artificial intelligence, we firstly developed a strategy for data preparation including encoding molecules with canonical molecular signatures and vectorizing bond-substrings by an embedding algorithm. Then, the dynamic neural network named Tree-LSTM is employed to depict molecular tree data-structures while the BPNN is used to correlate properties. To evaluate the performance of proposed DNN, the critical properties of nearly 1800 compounds are employed for training and testing the DNN models. As compared with classical group contribution methods, it can be demonstrated that the learned DNN models are able to provide more accurate prediction and cover more diverse molecular structures without considering frequencies of substructures.

Keywords: deep learning, neural network, signature molecular descriptor, property prediction, critical properties

Introduction

The chemical process and product design rely heavily on physical properties (*e.g.*, critical properties) and prediction models.[1,2] To investigate relationships between molecular structures and properties, plenty of mathematical models have been developed.[3] Most prediction models are based on semi-empirical quantitative structure property relationships (QSPRs) including group contribution (GC) methods and topological indices (TIs).

In GC methods, any compound can be divided into fragments (*e.g.*, atom, bond, group containing atoms and bonds). Each fragment has a partial value called a contribution, and the final property value is given by summing the fragmental contributions. A large variety of these models has been designed differing in the field of their applicability and in the set of experimental data. For example, GC methods reported by Lydersen,[4] Klincewicz and Reid,[5] Joback and Reid,[6] Constantinou and Gani,[7] and Marrero and Gani[8] are generally suitable to obtain values of physical properties, because these methods provide the advantage of quick estimation without substantial computational work. As alternative approaches, TIs, were used to estimate properties similar to the way of GC methods. In topological index (TI) methods, molecular topology is characterized depending on standard molecular graph properties such as vertex degrees, connectivity, atomic types, *etc.* Additionally, one of the main advantages is that TI methods can make a distinction between two similar structures from a more holistic perspective than GC methods.[9]

Another method named signature molecular descriptor that combining the advantages of

GC and TI methods was developed by Faulon *et al.*[10,11] Similar to TI methods, chemical structures is conceived of as chemical graphs. The signature descriptor retains all the structural and connectivity information of every atom in a molecule, rather than ascribe various numerical values to a complete molecular graph.[9] Meanwhile, the signature descriptor has ability to represent molecular substructures similar to GC methods. Faulon *et al*.[12] also introduced a canonical form of molecular signatures to solve molecular graph isomorphism which provides a holistic picture depicting molecular graphs and also holds the sub-structural information of a molecule. Nevertheless, we found the previous researches have few attempts to use the canonical molecular signature for QSPR modeling. To the best of our knowledge, the main reason is that the canonical molecular signature is not represented in a numeric form and it cannot be employed within the common-used mathematical models for QSPRs.

For the property estimation, most above-mentioned QSPR models, based on the specific rules such as a certain set of molecular substructures or an array of molecular graph-theoretic properties, are often formulated by multiple linear regressions (MLRs). Facts proved the MLR techniques have strong ability to correlate QSPRs, however, their encoding rules and mapping functions are defined a priori (*i.e.*, mathematical formulations are not adaptive to the different regression tasks). Moreover, the MLRs cannot be applied with the canonical molecular signatures for QSPR modeling. On the other hand, an alternative technique, the neural network, has been used to learn molecular structures and correlate physical properties or activities.[13] A variety of molecular descriptors (*e.g.*, topological characteristics, frequency of molecule substructures, and microscopic data of molecules) are fed to artificial neural networks. With the limitation of the computing capability and development platform at that

period, most researchers adopted feedforward neural networks with static computing graphs in their studies.[14-32]

Although these methods are well-used or precise in properties prediction, the molecular features are chosen manually as the input for above-mentioned models. For example, the splitting rules of molecular groups are pre-determined manually in the GC methods, or the well-chosen descriptors are input to the artificial neural networks (ANNs). With the number of various properties and product designs has been increasing, some properties/activities may need to be correlated with more molecular features or calculated by more complex mathematical models. It is therefore a challenge to pick out relevant features of molecules from massive data in the classical QSPR modeling.

Recently, many researchers were encouraged to study deep learning in artificial intelligence with improvements of computing performance. The deep learning is a much more intelligent technique that can capture the valuable features automatically. This advantage enables deep neural networks (DNNs) to formulate models from a great variety of big data. As such, some new information carriers (*e.g.*, graphs, images, texts, and 3D models) could be used to represent molecular structures in the QSPR modeling with DNNs. Lusci *et al.*[33] utilized the recurrent neural networks (RNNs) to present a molecular graph by considering molecules as undirected graphs and proposed an approach for mapping aqueous solubility to molecular structures. Goh *et al.*[34] developed a deep RNN "SMILES2vec" that automatically learns features from simplified molecular-input line-entry system[35] (SMILES) to correlate properties without the aid of additional explicit feature engineering. Goh *et al.*[36] also developed a deep convolutional neural network for the prediction of chemical properties,

using just the images of 2D drawings of molecules without providing any additional explicit chemistry knowledge such as periodicity, molecular descriptors, and fingerprints. These creative works[34,36] demonstrate the plausibility of using DNNs to assist in computational chemistry researches. The neural networks based on the long short-term memory (LSTM) units suggested by Hochreiter *et al*.[37] also have been adopted in the quantitative structure-activity relationship (QSAR) researches. Altae-Tran *et al*.[38] proposed a new deep learning architecture based on the iterative refinement LSTM to improve learning of meaningful distance metrics over small-molecules. The Tree-structure LSTM (Tree-LSTM) introduced by Tai *et al*.[39] is able to capture the syntactic properties of natural languages and two natural extensions were proposed depending on the basic LSTM architecture, which outperform other RNNs in their experiments. We noticed that the new neural network Tree-LSTM might be possible to depict the canonical molecular signature.

Motivated by the preceding researches, in this contribution, we focus on developing a deep learning approach that can learn QSPRs automatically and cover a wider range of substances for better predictive capabilities. A Python-based implementation with Faulon's algorithm[12] is achieved to convert molecules into canonical signatures for depicting molecular graphs and an in-house encoding approach is developed to parse the signatures into tree data-structures conveniently. The Tree-LSTM network and back-propagation neural network (BPNN) are incorporated into the DNN for modeling QSPR, among which, the Tree-LSTM mimics the tree structures of canonical signatures and outputs a feature vector that is used to correlate properties within a BPNN. As such, there is no need to convert molecules to bitmap images for training convolutional neural networks and to treat molecules as linear languages

for training RNNs. Then, the novelty of the proposed approach is that the canonical molecular signatures are used as templates to generate the topological structures of Tree-LSTM networks. In this sense, the contribution of this study is to propose an intelligent strategy of QSPR modeling based on deep learning that can extract the valuable features from molecular structures automatically. An important type of properties in process and product designs, critical properties, is used as case studies to clarify the main details of the deep learning architecture, which highlights the outperformance of the implemented QSPR modeling strategies within the proposed DNN.

Methodology

In this section, the technical details with respected to the deep learning architecture for modeling QSPR will be introduced. The proposed deep learning architecture incorporates multiple techniques that including canonical molecular signatures, word embedding, Tree-LSTM network, BPNN, *etc*. The proposed architecture consisting of eight steps is illustrated in Figure 1. Step 1 mainly involves the data acquisition of molecular structures, where the SMILES expressions are captured from open access databases. The second step is the embedding stage, where the vectors representing the substrings of chemical bonds are generated and collected into a dictionary with a widely used word-embedding algorithm. The third step is focused on the canonization of a molecule, where the molecular structures are transformed into the canonical molecular signatures as the templates for formulating the Tree-LSTM network. Step 4 refers to the mapping stage, where the adaptive structure of the Tree-LSTM network is obtained by the recursive algorithm from the canonical signature. In other words, the Tree-LSTM network is self-adaptive to a molecule. Step 5 involves the

inputting vectors of each substrings corresponding to each node. The Tree-LSTM network will be calculated from the lowest leaf node to the root node in this step. Finally, a vector representing a molecule is given from the root node. Step 6 is focused on the correlation stage of a property, where the vector representing a molecule is input into a BPNN to compute a scalar output for the property prediction. Step 7 is the comparison stage, where the tolerance between the predicted value and the experimental value is calculated. Step 8 is the feedback stage, where the adjustable parameters in the Tree-LSTM network and the BPNN are corrected for reducing the tolerance in step 7. The training process of the proposed DNN is the iterative loop within steps 5, 6, 7 and 8.

The Signature Molecular Descriptor

The canonical molecular signature is employed to depict molecules in this work. One reason is that a computer program can generate signatures automatically. Another important reason is that the canonical molecular signature provides a method to distinguish molecular structures for isomorphism. This also transforms the molecules with a uniform form for mapping to the neural network model.

To introduce canonical molecular signatures, atomic and molecular signatures have to be defined. An atomic signature is a subgraph originated at a specific root atom, and includes all atoms/bonds extending out to the predefined distance, without backtracking. The predefined distance is a user-specified parameter called the signature height *h*, and it determines the size of the local neighborhood of atoms in a molecule. It means that specified a certain root atom in a chemical graph, its atomic signature represents all of the atoms that are within a certain distance *h*, from the root. The atomic signature of atom *x* in height *h* given as ${}^{h}\sigma_{G(x)}$, is a

representation of the subgraph of the 2D graph G = (V, E) containing all atoms that are at distance h from x. It is noted that V and E correspond to the vertex (atom) set and edge (bond) set, respectively. Acetaldoxime (CAS No. 107-29-9) is taken as an example to provide atomic signatures shown in Figure S1 of Supporting Information. The carbon atom numbered by 0 (C0) is given as the root atom, and it is single-bonded to three hydrogen atoms and another carbon atom numbered by 1 (C1). Thus, the atomic signature for this root atom at height 1 is [C]([C][H][H][H]), the other atomic signatures are shown in Figure S1b.

In Faulon's theory,[8] the molecular signature shown in Figure S1c is a linear combination of all the atomic signatures, and is defined as Eq. 1.

$$^{h}\sigma(G) = \sum_{x \in V} \sigma_G(x) \quad (1)$$

In a given compound, any atomic signature can appear more than once. For example, the atomic signature [H]([C]) occurs four times in acetaldoxime. When the height of atomic signatures reaches the maximum value, the molecular graph can be reconstructed from any of the atomic signatures. Consequently, as long as graph canonization is concerned, there is no need to record all atomic signatures. The lexicographically largest atomic signature suffices to represent the graph in a unique manner.[10] For example, acetaldoxime has nine atomic signatures at the maximum height as shown in Figure S1d, and each of them is able to describe the complete molecular structure. If these nine signatures are sorted in decreasing lexicographic order (a canonical order), the lexicographically largest one can be defined as the canonical molecular signature that could be encoded and then mapped to the Tree-LSTM network.

Data Preparation: Molecules Encoding and Canonizing

In this work, SMILES expressions that used for depicting molecular structures are gathered from PubChem database.[40] We developed a program based on RDKit[41] for parsing and preserving the canonical molecular signature. The program implements Faulon's algorithm to generate and canonize atomic signatures, which can translate SMILES expressions to molecular graphs before canonizing molecular structures. There exist two rules for coding molecular structures in this program, one is the canonical string encoding a canonical molecular signature, and the other is the developed in-house coding method. The canonical molecular signature is used to determine the root atom in different molecules. However, it is difficult to reproduce the molecule structures and feed into the neural network from a molecular signature represented by a canonical string. When training the neural networks, one needs a more straightforward and simpler expression for parsing a molecule as a tree data-structure. As such, we developed a specified in-house coding method detailed in Supporting Information.

Data Preparation: Atom Embedding from Chemical Bonds

As the inputs of Tree-LSTM networks, atoms and bonds need to be translated and represented in form of vectors. Word embedding has been widely applied in natural language processing, several known program in the field has been developed, such as "Word2vec".[42] Inspired by this method, we proposed a simple approach to generate vector representations of atoms (see Figure 2) by breaking a chemical bond string into two smaller particles.

As we all know, chemical bonds are frequently represented in form of "A-B", "A" and "B" represent atoms, and "-" represents chemical bond types between two atoms. The string

as “A-B” is extracted from a data set of molecular structures, and then it is split into two part, “A” and “-B”, as the samples to train the embedding neural network. For this application, the skip-gram algorithm[42] is employed. As such, the substrings “A” and “-B” can be mapped into vectors for expressing each node in the Tree-LSTM network. In other words, a molecule is considered as a sentence in the embedding algorithm, and “A” or “-B” is equivalent to a word.

Here, the methane molecule including five atoms is taken as an example shown in Figure 2. Every atom is considered as the starting point to record its connected bonds and atoms. A dictionary is extracted from the samples of chemical bonds. The substrings “A” and “-B” are represented by some initial vectors, for example, one-hot codes. Each initial vector is employed to train the embedding neural network. Based on these training samples, the neural network will output probabilities representing that each substring of the dictionary is the next substring. After training completed, the weights of neurons in the embedding network are formed into target vectors.

Deep Neural Network

A DNN combining Tree-LSTM and BPNN is developed in this work. The Tree-LSTM neural network is employed for depicting molecular tree data-structures with the canonical molecular signatures while the BPNN is used to correlate properties.

The Child-sum Tree-LSTM can be used to the dependency tree while the N-ary Tree-LSTM is applied to the constituency tree,[39] and the mathematical models of these two Tree-LSTM models are listed in Table 1. The gating vectors and memory cell updates of the Tree-LSTM are dependent on the states of child units, which is different from the standard LSTM. Additionally, instead of a single forget gate, the Tree-LSTM unit contains one forget

gate f_{jk} for each child k. This allows the Tree-LSTM to incorporate information selectively from each child. Since the components of the Child-Sum Tree-LSTM unit are calculated from the sum of child hidden states h_k, the Child-Sum Tree-LSTM is well suited for trees with high branching factor or whose children are unordered. The vector $\widetilde{h}_j$ is the sum of the hidden states of all sub nodes under the current node j in the Child-sum Tree-LSTM model. The N-ary Tree-LSTM model can be utilized in the tree structure where the branching factor is at most N and where children are ordered from 1 to N. For any node j, the hidden state and memory cell of its k^{th} child are written as h_{jk} and c_{jk}, respectively. The introduction of separate parameter matrices for each child k allows the N-ary Tree-LSTM model to learn more fine-grained conditioning on the states of a unit's children than those of Child-Sum Tree-LSTM.

The performance evaluation of two Tree-LSTM models on semantic classification indicated that both Tree-LSTM models are superior to the sequential LSTM model and is able to provide better classification capability.[39] Therefore, the N-ary Tree-LSTM network is employed in this work to depict molecules, and the input variables are vectors converted by the embedding algorithm. In the QSPR model, the variable x_j is the input vector representing a substring of a bond ("A" or "-B"), and the vector h_j is the output vector representing a molecular structure. The vector h_j is finally associated with the properties by the BPNN. The BPNN involves an input layer, a hidden layer and an output layer. For other variables and functions in Table 1, $W^{(i,o,u,f)}$, $U^{(i,o,u,f)}$, $b^{(i,o,u,f)}$ are parameters that need to be learned, and σ represents the activation function sigmoid. For example, the model can learn parameters $W^{(i)}$ such that the components of the input gate i_j have values close to 1 (*i.e.*, "open") when an

important atom is given as input, and values close to 0 (*i.e.*, "closed") when the input is a less important atom. Taking acetaldoxime as an example again, the computing graph of the neural network is presented in Figure S3 of Supporting Information. It can be observed that the Tree-LSTM network mimics the topological structure of the acetaldoxime molecule. That is, if other molecular structures are learned, the Tree-LSTM network can vary the computing graph automatically. The BPNN accepts the output vectors from the Tree-LSTM network and correlates them with the property values. In this way, a DNN is built based on the Tree-LSTM network and BPNN.

Moreover, in this study, the aim of the DNN is to predict a numeric value instead of classification. Hereby, there is no need to employ the activation function "softmax".[43] The regularization technique "dropout"[44] is introduced to the BPNN for reducing overfitting. Huber loss[45] is adopted as the loss function in the training process, which is different from the frequently used classification scheme of Tree-LSTM network. More information about the DNN is disclosed in the section S4 of Supporting Information.

Model Training and Evaluation

The Tree-LSTM network has a dynamic computational graph that is a mutable directed graph with operations as vertices and data as edges. Hence, this neural network is implemented and trained in the deep learning framework PyTorch.[46] The Adam algorithm[47] is employed to train the DNN with a learning rate of 0.02 for the first 200 epochs, and subsequent epochs with 0.0001 in learning rate. Early stopping and batch normalization are utilized to decrease overfitting. The training process proceeded by monitoring the loss of test set and it will not finish until there is no improvement in the testing loss within continuous 50

epochs. Finally, the model with the lowest testing loss will be saved as the final model. To evaluate the correlative and predictive capacities of proposed deep learning architecture, the critical properties of pure compounds are adopted as case studies. It is acceptable that critical properties play vital roles in predicting phase behavior; however, the experimental measurements of critical properties are time-consuming, costly, and tough especially for large molecules that are easily decomposed. Moreover, several frequently used methods for the estimation of critical properties can be employed to compare with the learned DNN model. The values of critical properties are sourced from the Yaws' handbook[48] and the molecular structures of the relevant substances are gathered from PubChem database.[40] Several statistical metrics described in Supporting Information are used to evaluate the results of correlation and prediction for the DNN model.

Results and Discussion

The embedding vectors representing the bond-substrings are presented at first. The DNN's capability of correlation and prediction on the data set of critical properties is evaluated in the section, and it is compared with two classical GC methods (*i.e.*, Joback and Reid (JR) method, and Constantinou and Gani (CG) method).

Embedding of Bond Substrings

The input vectors of the Tree-LSTM network are translated from the substrings of the chemical bonds by the embedding neural network. After training, 106 substrings are extracted from the chemical bonds of 11052 molecules, and then they are converted to 50-dimensional real-valued vectors as the input data representing substring of every node in the Tree-LSTM network. This is contrasted to the more dimensions required for sparse word representations,

such as a one-hot encoding. These 50-dimensional vectors have been reduced to two dimensions by t-SNE algorithm[49] (see Figure S5 of Supporting Information) for understanding easily. More information about the embedding vectors is disclosed in the section S6 of Supporting Information.

The DNN Performance

The key idea behind the new deep learning architecture is to distinguish molecular structures by signature descriptors and to simulate molecule structures by a Tree-LSTM network. The QSPR models are obtained by training the DNN. The substances in the training and test sets are not screened carefully, which contains several small molecules and inorganic acids. Actually, these substances should be excluded from the modeling of the group contribution method, because they may cause deviation in the prediction. Finally, they are kept as some noise to the DNN. The predicting capabilities of the learned models are validated by a test set including independent compounds never used in training. The results of training and testing demonstrate that the Tree-LSTM network is capable of correlating physical properties and molecular structures (see Table 2).

The distributions of the standard deviation, average absolute error and average relative error are presented in Table 2 for three critical properties of training and test sets respectively. The number of data points for the average relative error that is less than 5% and greater than 10% are also presented. The residuals ($x^{exp} - x^{pred}$) of data points are plotted in the form of residual distribution plots in Figure 3. Also, the predicted values of these compounds by the proposed DNN in comparison with the experimental data are shown in Figure 4.

Comparisons with Existing Methods

Taken as examples, two existing GC methods for the estimation of critical properties are compared with the proposed DNN method, which involves JR method and CG method. The available performance data is provided by Poling *et al.*[50] We have to admit that the completely equitable comparison with other existing methods of property predictions is difficult since every method might be regressed from different data sources.

For the critical temperature (see Table 3), the JR method based on the experimental boiling points exhibits more accuracy than other GC methods, however, the accuracy of the JR method based on the estimated boiling points shows a marked decline.[50] To make the comparison as fair as possible, the substances from the same list provided by Poling *et al.*[50] are chosen to predict the critical temperature (T_c) using the proposed DNN. It can be seen from Table 3 that the DNN shows better performance than JR method (Est. Tb). It is noticed that the CG method involves groups in two orders, and the second order partially overcomes the limitation of the single order that cannot distinguish special molecular structures. Hereby, the number of substances estimated by the CG (2nd) method shown in Table 3 is actually a part from the substances estimated by the CG (1st) method. Although the list of 335 compounds within the CG method is not ascertained, it can be concluded that the accuracy of the learned DNN model is close to the CG method. When the learned DNN model is evaluated with all substances in the list provided by Poling *et al.*,[50] a decline in precision can be observed but the resulting *ARE* is still close to the others. Hereby, the DNN method can predict some substances that these GC methods cannot estimate, and the accuracy is close to the CG method for the critical temperature when the amount of substances engaged in the

comparison is approximate. Moreover, the DNN method also provides better precision when only predicting molecules with more than three carbon atoms.

For critical pressure (P_c), the estimations with the learned DNN model are more accurate than all other methods (see Table 4). It also proves that the method can correlate properties with more substances and has better accuracy for predicting the critical pressure. Furthermore, for the estimation of the critical volume (V_c), as indicated in Table 5, the estimation of the critical volume with the DNN method reaches precision close to other methods.

Actually, in total 468 substances are provided by Poling *et al*.[50] It can be observed that the number of substances estimated by the CG method and JR method is less than the learned DNN model. In other words, the critical properties of some substances cannot be predicted using these two existing GC methods. The reason is that the GC methods are limited by the types and segmentation rules of groups while the DNN method is not subject to them. Hereby, the DNN can predict more compounds and achieve a decent precision while it performs the acceptable precision on the substances provided by Poling *et al.*[50]

Another important fact has also to be considered is the compounds exemplified in Tables 3-5 have been involved more or less in the regression samples of the JR method, the CG method and the DNN. Although the above-mentioned comparison can evaluate the predictive capability of the DNN differed from those two existing GC methods, the extrapolation ability is also necessary to be evaluated. There is no program available for us to estimate properties by the CG method, Figure 5 only shows the comparison of extrapolation abilities between the JR method and the DNN according to the substances of test sets shown in Table 3. More details are described in the section S7 of Supporting Information, and the comparison between

a reported neural network-based method and the learned DNN model is exhibited.

Distinction of isomers

Signature descriptors have the ability to distinguish isomers. Table 6 exhibits the estimations through the DNN method as opposed to experimental values and other GC methods. Apparently, the JR method cannot recognize isomers, although it is able to predict more accurate according to the experimental boiling point. The CG method with the second order of groups can obtain decent prediction for isomers, and the DNN method can achieve similar results.

Conclusions

In this work, a deep learning architecture is developed and the prediction of physical properties from the holistic molecule structure is achieved in following four steps. Firstly, an embedding neural network is used to generate the vector representations of bond-substrings. Then, a canonization algorithm is employed to convert the molecules to uniform data-structures for providing templates to the Tree-LSTM neural network. Next, the computational graph of the Tree-LSTM network accepts a vector of bond-string on its each node, which is self-adaptive to various molecular structures. Finally, a vector outputting from the root node of the Tree-LSTM network is introduced to a BPNN to generate predictive property values. The proposed DNN does not rely on the well-chosen descriptors to correlate properties, it could learn some valuable features of molecule and achieve an acceptable precision of a specific property for more substances with less human effort.

The proposed approach neither counts the frequencies of molecular substructures nor calculates any numerical descriptors, instead, provides a way to build the QSPR models from

the text-type descriptor, the canonical signature representing molecular graphs. Hence, the strategy has a capability to capture the relevant molecular features for QSPR modeling automatically. Furthermore, those parameters involved in the learned DNN model are not the contribution value of each group in GC methods but tensors containing potential information.

For validating the effectiveness of the proposed deep learning architecture, critical properties are taken as case studies to train and test the QSPR models built from the proposed DNN combining Tree-LSTM and back-propagation neural network (BPNN). It has been proven that these QSPR models provide more accurate prediction and cover more diverse molecular structures. Moreover, the DNN behaves a better ability in distinguishing isomers. We admit that the data used to train the model is still far from enough. This signifies that there needs to be more data to capture the delicate relationships that may exist between molecule structures and physical properties.

In a word, the wide applicability of the proposed architecture highlights the significance of deep learning providing an intelligent tool to predict properties in the design or synthesis of chemical, pharmaceutical, bio-chemical products and processes. It is worth mentioning that the proposed strategy could be widely applied for the estimation of other properties of pure compounds, such as environment-related properties and safety-related properties.

Acknowledgement

We acknowledge the financial support provided by the National Natural Science Foundation of China (Nos. 21878028, 21606026); the Fundamental Research Funds for the Central Universities (No. 2019CDQYHG021); the Chongqing Research Program of Basic Research and Frontier Technology (No. CSTC2016JCYJA0474).

Literatures Cited

1 Shen WF, Dong LC, Wei SA, Li J, Benyounes H, You XQ, Gerbaud V. Systematic design of an extractive distillation for maximum-boiling azeotropes with heavy entrainers. *AIChE J*. 2015;61(11):3898-3910.

2 Yang A, Shen WF, Wei SA, Dong LC, Li J, Gerbaud V. Design and control of pressure-swing distillation for separating ternary systems with three binary minimum azeotropes. *AIChE J.* 2019;65(4):1281-1293.

3 Kontogeorgis GM, Gani R. Introduction to computer aided property estimation. In: Kontogeorgis GM, Gani R. Computer aided property estimation for process and product design. Amsterdam, the Netherlands: Elsevier; 2004;3-26.

4 Lydersen AL. Estimation of critical properties of organic compounds. University of Wisconsin College of Engineering. *Eng Exp Stn Rep*. Madison: WI; 1995.

5 Klincewicz KM, Reid RC. Estimation of critical properties with group contribution methods. *AIChE J*. 1984;30(1):137-142.

6 Joback KG, Reid RC. Estimation of pure-component properties from group-contributions. *Chem Eng Commun*. 1987;57(1-6):233-243.

7 Constantinou L, Gani R. New group contribution method for estimating properties of pure compounds. *AIChE J*. 1994;40(10):1697-1710.

8 Marrero J, Gani R. Group-contribution based estimation of pure component properties. *Fluid Phase Equilib*. 2001;183-184:183-208.

9 Austin ND, Sahinidis NV, Trahan DW. Computer-aided molecular design: An introduction and review of tools, applications, and solution techniques. *Chem Eng Res*

Des. 2016;116:2-26.

10 Faulon JL, Visco DP, Pophale RS. The signature molecular descriptor. 1. Using extended valence sequences in QSAR and QSPR studies. *J Chem Inf Comput Sci*. 2003;43(3):707-720.

11 Faulon JL, Churchwell CJ, Jr VD. The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences. *J Chem Inf Comput Sci*. 2003;34(3):721-734.

12 Faulon JL, Collins MJ, Carr RD. The signature molecular descriptor. 4. Canonizing molecules using extended valence sequences. *J Chem Inf Comput Sci*. 2004;44(2):427-436.

13 Borman S. Neural network applications in chemistry begin to appear. *Chem Eng News*. 1989;67(17):24-28.

14 Bodor N, Harget A, Huang MJ. Neural network studies. 1. Estimation of the aqueous solubility of organic compounds. *J Am Chem Soc*. 1991;113(25):9480-9483.

15 Aoyama T, Suzuki Y, Ichikawa H. Neural networks applied to structure-activity relationships. *J Med Chem*. 1990;33(3):905-908.

16 Egolf LM, Jurs PC. Prediction of boiling points of organic heterocyclic compounds using regression and neural network techniques. *J Chem Inf Comput Sci*. 1993;33(4):616-625.

17 Kireev DB. ChemNet: A novel neural network based method for graph/property mapping. *J Chem Inf Comput Sci*. 1995;35(2):175-180.

18 Devillers J. Neural networks in QSAR and drug design, principles of QSAR and drug design. 2nd ed. San Diego: Academic Press; 1996.

19 Bünz AP, Braun B, Janowsky R. Application of quantitative structure-performance relationship and neural network models for the prediction of physical properties from molecular structure. *Ind Eng Chem Res*. 1998;37(8):3043-3051.

20 Beck B, Breindl A, Clark T. QM/NN QSPR models with error estimation: vapor pressure and logP. *J Chem Inf Comput Sci*. 2000;40(4):1046-1051.

21 Espinosa G, Yaffe D, Cohen Y, Arenas A, Giralt F. Neural network based quantitative structural property relations (QSPRs) for predicting boiling points of aliphatic hydrocarbons. *J Chem Inf Comput Sci*. 2000;40(3):859-879.20

22 Yao X, Zhang X, Zhang R, Liu M, Hu Z, Fan B. Prediction of enthalpy of alkanes by the use of radial basis function neural networks. *Comput Chem.* 2001;25(5):475-482.

23 Yaffe D, Cohen Y. Neural network based temperature-dependent quantitative structure property relations (QSPRs) for predicting vapor pressure of hydrocarbons. *J Chem Inf Comput Sci*. 2001;41(2):463-477.

24 Yaffe D, Cohen Y, Espinosa G, Arenas A, Giralt F. Fuzzy ARTMAP and back-propagation neural networks based quantitative structure-property relationships (QSPRs) for octanol-water partition coefficient of organic compounds. *J Chem Inf Comput Sci*. 2002;42(2):162-183.

25 And TAA, George RS. Artificial neural network investigation of the structural group contribution method for predicting pure components auto ignition temperature. *Ind Eng Chem Res*. 2003;42(22):5708-5714.

26 Chiu TL, So SS. Development of neural network QSPR models for Hansch substituent constants. 1. Method and validations. *J Chem Inf Comput Sci*. 2004;44(1):147-153.

27 Torrecilla JS, Rodríguez F, Bravo JL, Rothenberg G, Seddon KR, Lopez-Martin I. Optimising an artificial neural network for predicting the melting point of ionic liquids. *Phys Chem Chem Phys*. 2008;10(38):5826-5831.

28 Gharagheizi F, Alamdari RF, Angaji MT. A new neural network-group contribution method for estimation of flash point temperature of pure components. *Energy Fuels*. 2008;22(3):1628-1635.

29 Gharagheizi F. New neural network group contribution model for estimation of lower flammability limit temperature of pure compounds. *Ind Eng Chem Res*. 2009;48(15):7406-7416.

30 Wang R, Jiang J, Pan Y, Cao H, Cui Y. Prediction of impact sensitivity of nitro energetic compounds by neural network based on electrotopological-state indices. *J Hazard Mater.* 2009;166:155-186.

31 Guerra A, Campillo NE, Páez JA. Neural computational prediction of oral drug absorption based on CODES 2D descriptors. *Eur J Med Chem.* 2010;45:930-940.

32 Bagheri M, Borhani TNG, Zahedi G. Estimation of flash point and autoignition temperature of organic sulfur chemicals. *Energy Convers Manage*. 2012;58:185-196.

33 Lusci A, Pollastri G, Baldi P. Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. *J Chem Inf Comput Sci*. 2013;53(7):1563-1575.

34 Goh GB, Hodas NO, Siegel C, Vishnu A. Smiles2vec: an interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint.* 2017;arXiv:1712.02034.

35 D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci*. 1988;28(1):31-36.

36 Goh GB, Siegel C, Vishnu A, Hodas NO, Baker N. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. *arXiv preprint.* 2017;arXiv:1706.06689.

37 Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.

38 Altae-Tran H, Ramsundar B, Pappu AS, Pande V. Low data drug discovery with one-shot learning. *ACS Cent Sci*. 2017;3(4):283-293.

39 Tai KS, Socher R, Manning CD. Improved semantic representations from tree-structured long short-term memory networks. *Comput Sci*. 2015;5(1):36.

40 Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, Han L, He J, He S, Shoemaker BA, Wang J, Yu B, Zhang J, Bryant SH. PubChem substance and compound databases. *Nucleic Acids Res*. 2015;44(D1):D1202-D1213.

41 Landrum G. Rdkit: Open-source cheminformatics software. 2017. Available at: http://www.rdkit.org/, https://github.com/rdkit/rdkit.

42 McCormick C. Word2Vec tutorial-the skip-gram model. 2016. Available at: http://www.mccormickml.com.

43 Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-normalizing neural networks. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan, S, Garnett R. Advances in Neural Information Processng Systems 30. New York: Curran Associates, Inc., 2017:971-980.

44 Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. *Comput Sci*. 2012;3(4):212-223.

45 Huber PJ. Robust estimation of a location parameter. *Ann Math Stat.* 1964;35(1):73-101.

46 Paszke A, Gross S, Chintala S, Chanan G. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. 2019. Available at: https://pytorch.org.

47 Kingma DP, Ba J. Adam: a method for stochastic optimization. *Comput Sci*. 2014.

48 Yaws CL, Gabbula C. Yaws' handbook of thermodynamic and physical properties of chemical compounds. New York: Knovel; 2003.

49 Maaten LVD, Hinton G. Visualizing data using t-sne. *J Mach Learn Res*, 2008; 9(2605): 2579-2605.

50 Poling BE, Prausnitz JM, O'connell JP. The properties of gases and liquids. 5th ed. New York: McGraw-Hill; 2001.

List of Figures Captions

Figure 1. Schematic diagram of technical architecture for deep learning in the prediction of physical properties.

Figure 2. The procedure of the embedding neural network for vectorizing the bond-strings.

Figure 3. Residual distribution plots for the training and test sets of: (a) T_c; (b) P_c; (c) V_c.

Figure 4. Plots of estimated versus experimental values for training and test sets of: (a) T_c; (b) P_c; (c) V_c.

Figure 5. Plots of estimated versus experimental values of critical properties (the DNN vs. the JR method) of: (a) T_c; (b) P_c; (c) V_c.

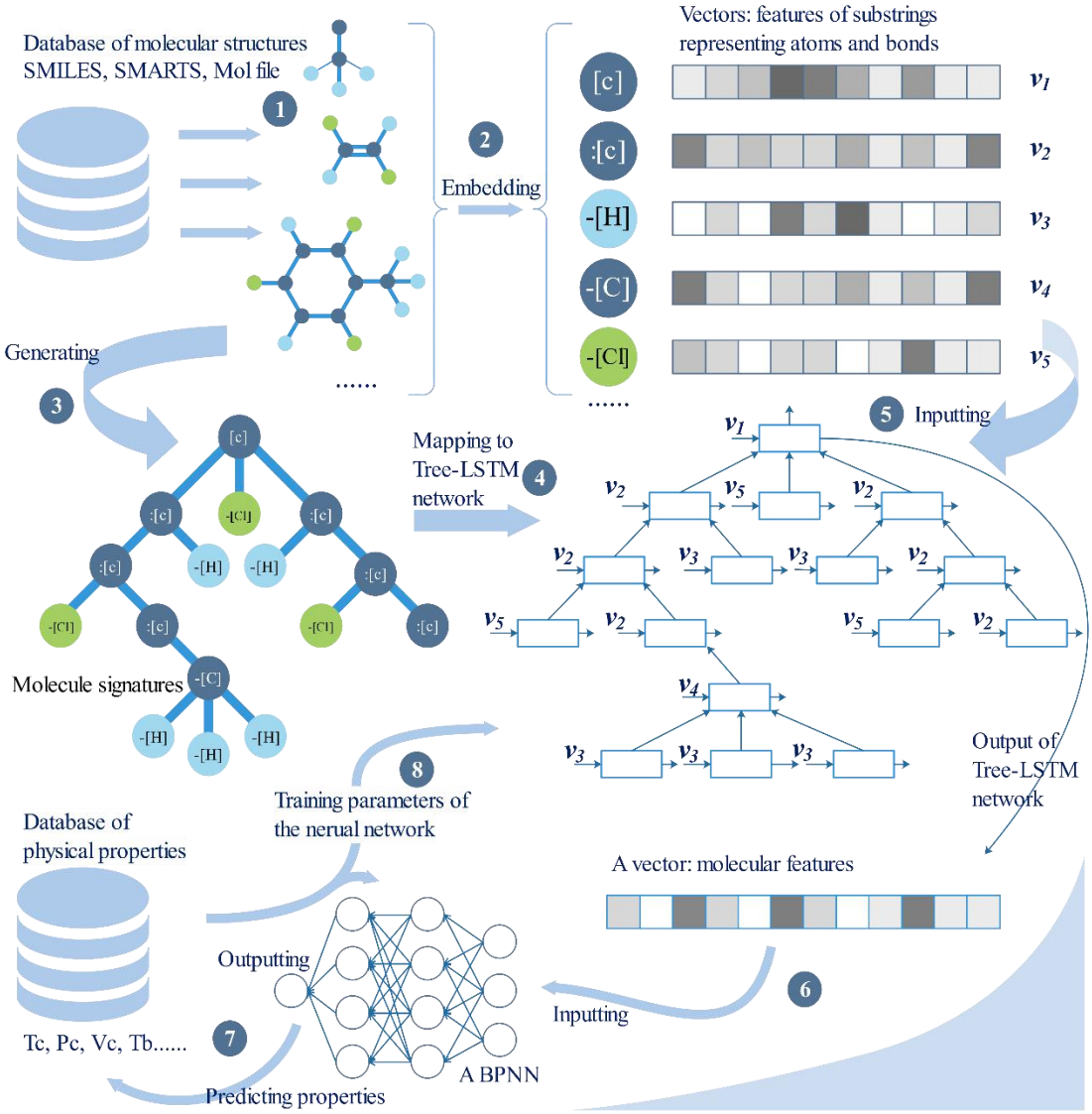

Figure 1. Schematic diagram of technical architecture for deep learning in the prediction of physical properties.

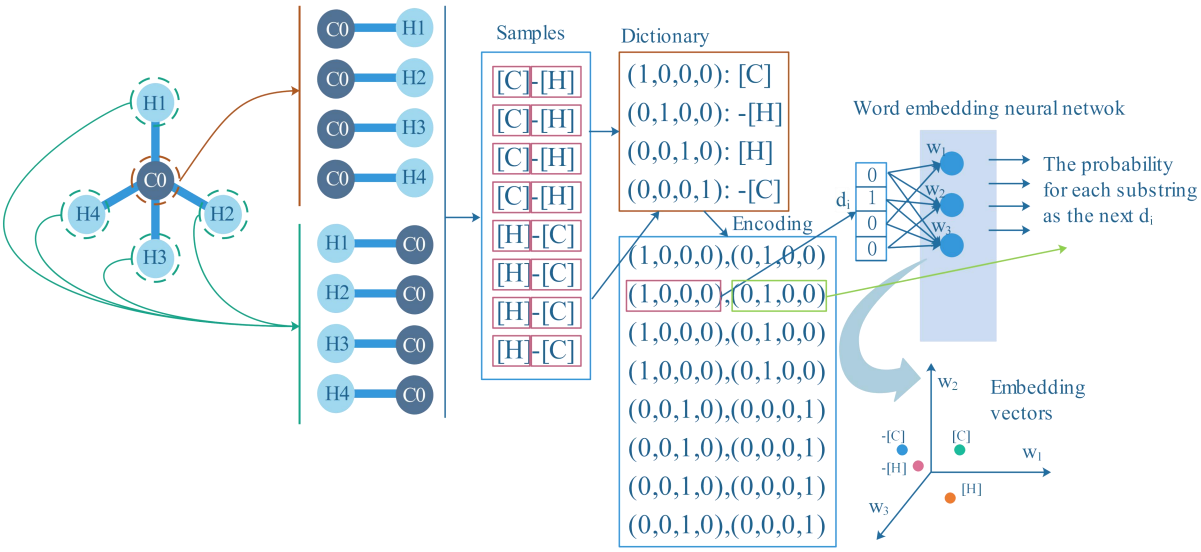

Figure 2. The procedure of the embedding neural network for vectorizing the bond-strings.

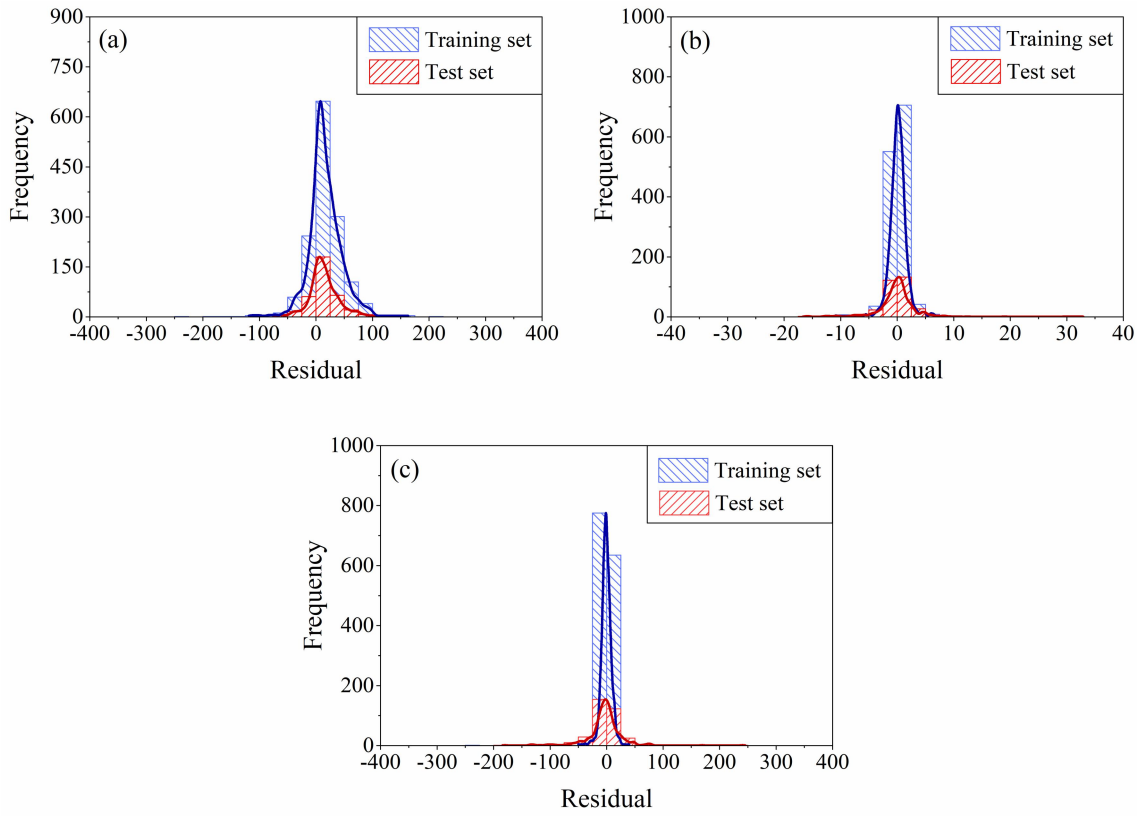

Figure 3. Residual distribution plots for the training and test sets of: (a) T_c; (b) P_c; (c) V_c.

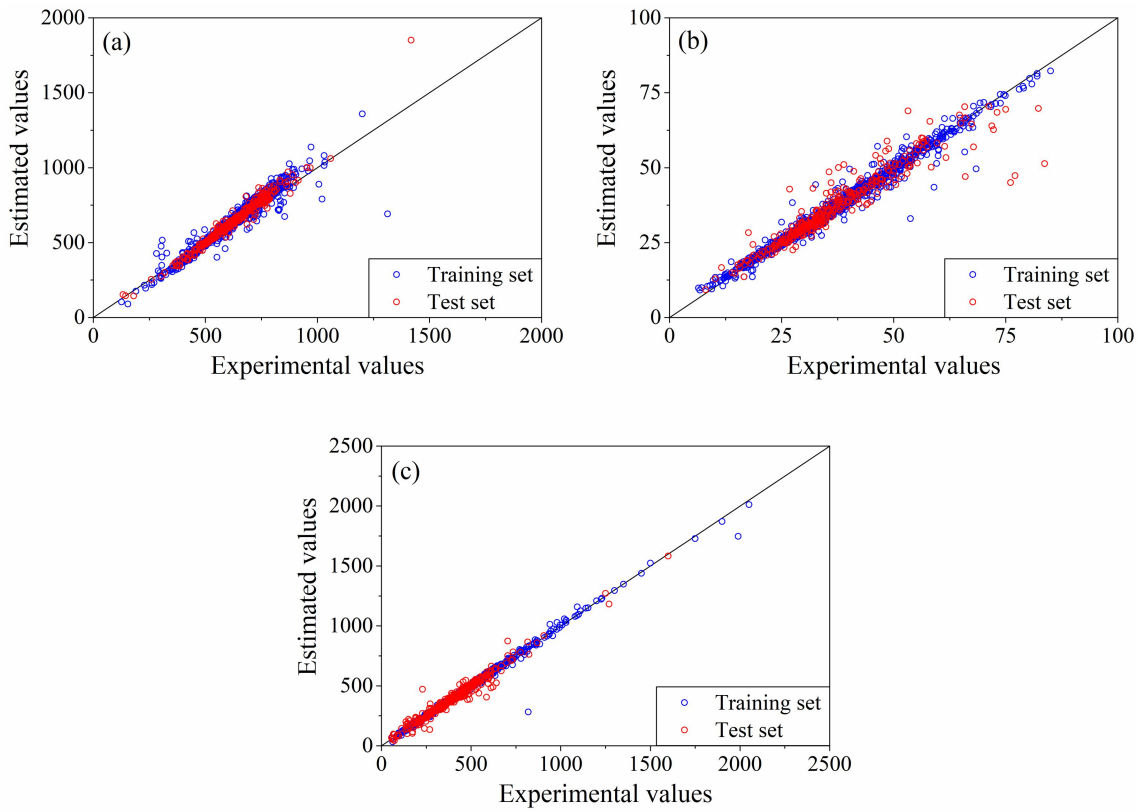

Figure 4. Plots of estimated versus experimental values for training and test sets of: (a) T_c; (b) P_c; (c) V_c.

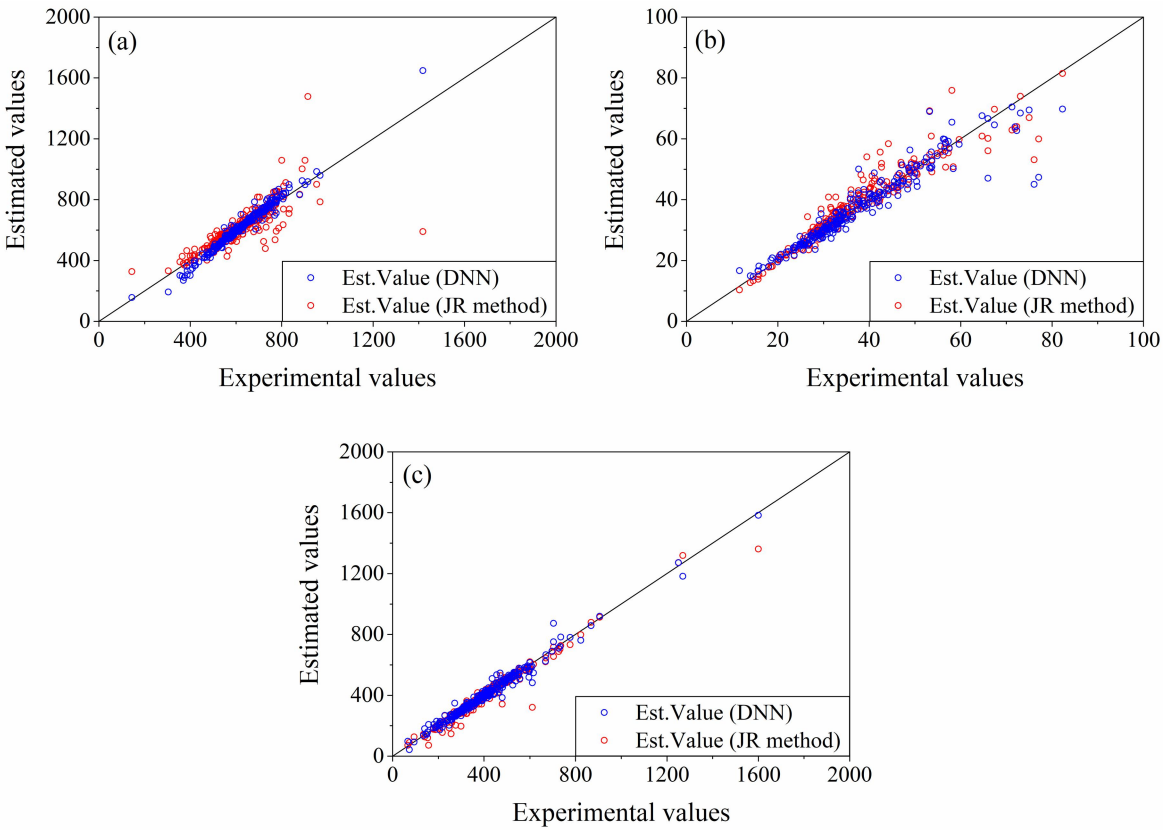

Figure 5. Plots of estimated versus experimental values of critical properties (the DNN vs. the JR method) of: (a) T_c; (b) P_c; (c) V_c.

Table 1. The transition equations of Child-sum Tree-LSTM and N-ary Tree-LSTM[39]

Child-sum Tree-LSTM		N-ary Tree-LSTM	
$\tilde{h}_j = \sum_{k \in C(j)} h_k$	(2)	- *	
$i_j = \sigma(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)})$	(3)	$i_j = \sigma(W^{(i)} x_j + \sum_{l=1}^{N} U_l^{(i)} h_{jl} + b^{(i)})$	(9)
$f_{jk} = \sigma(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}$	(4)	$f_{jk} = \sigma(W^{(f)} x_j + \sum_{l=1}^{N} U_{kl}^{(f)} h_{jl} + b^{(f)})$	(10)
$o_j = \sigma(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)})$	(5)	$o_j = \sigma(W^{(o)} x_j + \sum_{l=1}^{N} U_l^{(o)} h_{jl} + b^{(o)})$	(11)
$u_j = \tanh(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)})$	(6)	$u_j = \tanh(W^{(u)} x_j + \sum_{l=1}^{N} U_l^{(u)} h_{jl} + b^{(u)}$	(12)
$c_j = i_j \bullet u_j + \sum_{k \in C(j)} f_{jk} \bullet c_k$	(7)	$c_j = i_j \bullet u_j + \sum_{l=1}^{N} f_{jl} \bullet c_{jl}$	(13)
$h_j = o_j \bullet \tanh(c_j)$	(8)	$h_j = o_j \bullet \tanh(c_j)$	(14)

* "-" represents null since $\tilde{h}_j$ is not involved in the N-ary Tree-LSTM unit. Notations and symbols are described in the Supporting Information.

Table 2. Global comparison of critical properties between training and test sets

Properties	Data points		s [a]		AAE [b]		ARE (%) [c]		# Err < 5%		# Err >10%	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
T_c (K)	1432	360	145.89	166.09	22.48	23.77	4.23	5.29	1104	266	109	36
P_c ($\times 10^5$ Pa)	1380	346	161.80	139.55	1.34	3.18	3.81	8.29	1104	177	98	89
V_c ($\times 10^{-6}$ m^3/mol)	1440	361	199.18	169.73	7.10	19.92	1.97	6.15	1361	245	19	59

[a] s is standard deviation; [b] AAE is average absolute error; [c] ARE is average relative error.

Table 3. The comparisons among DNN and GC methods in predicting critical temperature

Methods	Substances	*AAE* [c]	*ARE* [c]	# Err<5% [d]	# Err>10% [e]
JR[48] (Exp. T_b) [f]	352 [a]	6.65	1.15	345	0
	290 [b]	6.68	1.10	286	0
JR[48] (Est. T_b) [g]	352 [a]	25.01	4.97	248	46
	290 [b]	20.19	3.49	229	18
DNN [h]	352 [a]	15.39	2.92	299	15
	290 [b]	13.92	2.31	265	7
CG (1st)[48]	335 [a]	18.48	3.74	273	28
	286 [b]	13.34	2.25	254	4
CG (2nd)[48]	108 [a]	17.69	13.61	274	29
	104 [b]	12.49	2.12	254	6
DNN [i]	452 [a]	26.59	5.87	343	51
	335 [b]	15.98	2.62	294	11

[a] The number of substances in the list provided by Poling *et al.*[50] with data that could be tested with the method in the current line.

[b] The number of substances in the list provided by Poling *et al.*[50] having three or more carbon atoms with data that could be tested with the method in the current line.

[c] *AAE* is average absolute error; *ARE* is average relative error.

[d] The number of substances for which the *ARE* was less than 5% (# Err<5%).

[e] The number of substances for which the *ARE* was greater than 10% (# Err>10%). The number of substances with errors between 5% and 10% can be determined from the table information.

[f] The values of estimation is based on the experimental values of normal boiling point.

[g] The values of estimation is based on the estimation values of normal boiling point.

[h] The number of substances is kept consistent with the JR method.

[i] The number of all the substances that could be predicted by DNN.

Table 4. The comparisons among DNN and GC methods in predicting critical pressure

Methods	Substances	*AAE* [c]	*ARE* [c]	# Err<5% [d]	# Err>10% [e]
JR[48]	328 [a]	2.19	5.94	196	59
	266 [b]	1.39	4.59	180	30
DNN [h]	328 [a]	1.46	4.03	248	23
	266 [b]	1.21	3.94	206	19
CG (1st)[48]	316 [a]	2.88	7.37	182	52
	263 [b]	1.80	5.50	156	32
CG (2nd)[48]	99 [a]	2.88	7.37	187	56
	96 [b]	1.80	5.50	160	36
DNN [i]	450 [a]	2.66	5.43	314	58
	335 [b]	1.33	4.40	241	26

Note: [a] The number of substances in the list provided by Poling *et al.*[50] with data that could be tested with the method in the current line.

[b] The number of substances in the list provided by Poling *et al.*[50] having 3 or more carbon atoms with data that could be tested with the method in the current line.

[c] *AAE* is average absolute error; *ARE* is average relative error.

[d] The number of substances for which the *ARE* was less than 5% (# Err<5%).

[e] The number of substances for which the *ARE* was greater than 10% (# Err>10%). The number of substances with errors between 5% and 10% can be determined from the table information.

[f] The values of estimation is based on the experimental values of normal boiling point.

[g] The values of estimation is based on the estimation values of normal boiling point.

[h] The number of substances is kept consistent with the JR method.

[i] The number of all the substances could be predicted by DNN.

Table 5. The comparisons among DNN and GC methods in predicting critical volume

Methods	Substances	*AAE* [c]	*ARE* [c]	# Err<5% [d]	# Err>10% [e]
JR[48]	236 [a]	12.53	3.37	189	13
	185 [b]	13.98	3.11	148	9
DNN [h]	236 [a]	10.07	2.99	197	13
	185 [b]	11.20	2.69	157	10
CG (1st)[48]	220 [a]	15.99	4.38	160	18
	180 [b]	16.68	4.57	159	22
CG (2nd)[48]	76 [a]	16.5	3.49	136	10
	72 [b]	17.4	3.70	134	15
DNN [i]	402 [a]	15.05	4.84	301	56
	230 [b]	17.38	4.20	236	31

Note: [a] The number of substances in the list provided by Poling *et al.*[50] with data that could be tested with the method in the current line.

[b] The number of substances in the list provided by Poling *et al.*[50] having 3 or more carbon atoms with data that could be tested with the method in the current line.

[c] *AAE* is average absolute error; *ARE* is average relative error.

[d] The number of substances for which the *ARE* was less than 5% (# Err<5%).

[e] The number of substances for which the *ARE* was greater than 10% (# Err>10%). The number of substances with errors between 5% and 10% can be determined from the table information.

[f] The values of estimation is based on the experimental values of normal boiling point.

[g] The values of estimation is based on the estimation values of normal boiling point.

[h] The number of substances is kept consistent with the JR method.

[i] The number of all the substances could be predicted by DNN.

Table 6. Experimental and estimated critical temperature values of isomeric trimethylpentane and methyl propanol

Compounds	CAS No.	Exp. Value (K)	JR method (Est. T_b) Est. Value (K)	JR method (Exp. T_b) Est. Value (K)	CG method Est. Value (K)	DNN Est. Value (K)
2,2,3-Trimethylpentane	564-02-3	563.40	557.09	563.31	562.10	563.42
2,2,4-Trimethylpentane	540-84-1	543.90	557.09	547.71	540.33	544.98
2,3,3-Trimethylpentane	560-21-4	573.50	557.09	570.55	577.45	576.88
2,3,4-Trimethylpentane	560-21-4	566.30	556.23	564.26	581.37	565.84
2-Methyl-1-propanol	78-83-1	547.78	548.34	546.11	543.32	552.96
2-Methyl-2-propanol	75-65-0	506.20	548.34	509.38	497.46	500.57