IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 29, 2020, accepted March 12, 2020, date of publication March 19, 2020, date of current version April 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981874

Computational Prediction of Critical Temperatures of Superconductors Based on Convolutional Gradient Boosting Decision Trees

YABO DAN[1], RONGZHI DONG[1], ZHUO CAO[1], XIANG LI[1], CHENGCHENG NIU[2], SHAOBO LI[1,2], AND JIANJUN HU[1,3]

[1]School of Mechanical Engineering, Guizhou University, Guiyang 550025, China
[2]Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang 550025, China
[3]Department of Computer Science and Engineering, University of South Carolina at Columbia, Columbia, SC 29208, USA

Corresponding authors: Shaobo Li (lishaobo@gzu.edu.cn) and Jianjun Hu (jianjunh@cse.sc.edu)

Research reported in this publication was partially supported by the NSF under Grant OIA-1655740 (via SC EPSCoR/IDeA GEAR-CRP2019 19-GC02) and 1905775. The views, perspective, and content do not necessarily represent the official views of the SC EPSCoR/IDeA Program nor those of the NSF. It was also partially supported by National Important Project 2018AAA0101803 and Guizhou Province Science and Technology Project [2015] 4011.

ABSTRACT Superconductors have been one of the most intriguing materials since they were discovered more than a century ago. However, superconductors at room temperature have yet to be discovered. On the other hand, machine learning and especially deep learning has been increasingly used in material properties prediction and discovery in recent years. In this paper, we propose to combine the deep convolutional neural network (CNN) model with fully convolutional layers for feature extraction with gradient boosting decision tree (GBDT) for superconductors critical temperature (Tc) prediction. Our prediction model only uses the elemental property statistics of the materials as original input and learns a hierarchical representation of superconductors using convolutional layers. Computational experiments showed that our convolutional gradient boosting decision tree (ConvGBDT) model achieved the state-of-the-art results on three superconductor data sets: DataS, DataH, and DataK. By visually comparing the raw elemental feature distribution and the learned feature distribution, it is found that the convolutional layers of our ConvGBDT can learn features that can more effectively distinguish cuprate and iron-based superconductors. On the other hand, the GBDT part of our ConvGBDT model can learn the sophisticated mapping relationship between extracted features and the critical temperatures to obtain good prediction performance.

INDEX TERMS Superconductivity, convolutional neural network, gradient boosting decision tree, feature extraction.

I. INTRODUCTION

With their unique physical properties, superconductors play a critical role in many fields such as medical instruments, transportation, cutting-edge scientific equipment, controllable nuclear fusion, and power systems [1]–[5]. Since discovered in 1911, superconductivity has been found on low-temperature superconductors represented by NbTi, Nb_3Sn, etc., the first-generation high-temperature superconductors represented by Bi-Sr-Ca-Cu-O, the second-generation high-temperature superconductors represented by Re-Ba-Cu-O, and the later discovered superconductors such as MgB2 and iron-based superconductors. As a special physical property of multi-particle systems, the mystery of its mechanism has not been fully understood despite that some heuristic rules have been found. But currently, it is largely impossible to accurately predict high-temperature superconductors.

Most superconductivity studies use resource-intensive experiments or first-principles calculations. However, experimental exploration of the huge materials space is costly prohibitive while atomic computational models, especially submicron-level computational simulation methods are limited by the lack of well-defined force fields to describe the interactions between atoms [6]–[9]. On the other hand, rigorous electronic structure calculations using density

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

functional theory (DFT) are usually limited to simulating hundreds of atoms [7]–[9]. Standard material computational characterization methods such as Green function (e.g. the fully adaptive GW) [10], [11] considers finite-scale scaling, charge correction [12], and beyond standard density functional theory (BSDFT) [10], [11] when calculating band structures, which makes simulation calculations very expensive. and high-throughput screening impractical [13]. Essentially, in addition to prediction models based on physical principles/theories, the machine learning approach for Tc prediction is a data-driven prediction model, which exploits the relationship between material composition similarity and Tc. This is why the machine learning method has been successfully applied to the Tc prediction problems using only the composition features.

Currently, the availability of an increasing number of materials databases such as Materials Project (MP) Database [14] and ICSD [15] with experimental and/or computational properties has led to the recent emergence of machine learning for materials property prediction. At the same time, as DFT provides a cheaper way to predict material properties at the atomic level [16], DFT calculation results have been deposited into large data collections, such as Open Quantum Materials Database (OQMD) [17], [18], Automatic Flow of Materials Discovery Library (AFLOWLIB) [19], and the Novel Materials Discovery Database (NoMaD) [20]. These materials databases contain 10^4-10^6 DFT computational properties of both experimentally observed and hypothesized materials. In the superconductor research field, the most comprehensive database is the Supercon database [21], which contains the compositions and the Tc of 30,057 oxides metallic or 514 organic superconductors as of April 17, 2019. In the past decade, these materials databases have been applied to data-driven material informatics researches [22]–[27]. Indeed, these large data sources have spurred researchers' interest in applying advanced data-driven machine learning techniques to accelerate the discovery and design of new materials with selected engineering attributes [28]–[30]. Following this strategy, Stanev *et al.* [31] recently used Magpie [32] descriptors to characterize superconductors into 132-dimensional vectors and used random forest (RF) algorithm to develop a Tc prediction model using 6196 materials with Tc greater than 10K from the SuperCon database. Hamidieh [33] used the same characterization method to establish a Tc prediction model using 21,263 superconducting materials from SuperCon using GBDT.

However, conventional machine learning algorithms suffer from the difficulty of hand-designing effective features for building high-performance prediction models. Recently, deep learning has brought the state-of-the-art performance to tasks in various fields, including image recognition [34], [35], speech recognition [36], [37] and natural language understanding [38], [39]. As one type of deep neural network models, CNN is good at learning hierarchical features from the original data. It was first successfully applied in computer vision and since then has achieved remarkable success in many other areas such as gear fault diagnosis [40], [41], in which vibration and sound signals are converted into matrix/image format to exploit CNN's feature learning capabilities. In the field of material research, CNN models have been used to improve the characterization method after modeling the microstructure data of the material [42]–[44] and to predict the crystal structure and molecular properties [45]–[47]. Following this strategy, Konno *et al.* [48] proposed a four-channel material characterization method based on the electron number of the element s, p, d, and f orbitals in the molecular formula of superconductors, and used CNN to construct a critical temperature prediction model with 13,000 superconducting materials from the SuperCon database.

CNN are characterized by their hierarchical feature extraction capability, which is usually linked with fully connected layers for regression or classification. However, in the case of small data sets, the fully connected layers cannot be too complicated. Otherwise, the model is prone to overfitting and does not give good results on the test set. If the fully connected layer is too simple, the final regression model cannot provide sufficient nonlinear transformations to capture the relationship between input features and the predictions. On the other hand, the GBDT algorithm is a robust ensemble prediction model with strong modeling performance. Here we propose the ConvGBDT regression model by combining the convolutional layers of CNN with GBDT, which has achieved the best prediction performance over three benchmark datasets. The contributions of this paper can be summarized as follows:

(1) A hierarchical feature extraction method based on CNNs is developed for Tc prediction of superconductors using the elemental property representation of the materials.

(2) Using the GBDT model to replace the fully connected layer in regular CNNs, we developed a more effective prediction model for predicting Tc of superconductors.

(3) Extensive computational tests over three standard benchmark datasets demonstrate the state-of-the-art performance of our proposed ConvGBDT model.

II. DATA SOURCE AND CHARACTERIZATION

A. DATA SOURCES

We use the data derived from the SuperCon [21] database to train and test the ConvGBDT model. The SuperCon database contains a comprehensive list of superconductors, all of which are collected from journals of published papers. SuperCon contains two kinds of compounds, one is the metal oxide (metal-containing inorganic material, alloy compound, oxide high-temperature superconductor, etc.), and the other is organic superconductors. The SuperCon dataset continues to evolve and at the time of writing contains 30,000+ superconductors that change only by small changes in stoichiometry (doping plays an important role in optimizing the Tc of superconducting materials). The SuperCond database has been used by many scholars in the construction of Tc prediction model including Stanev *et al.* [31], Hamidieh [33],

TABLE 1. Statistics of the three benchmark superconductor datasets.

	No. of materials	No.of cuprates	No.of iron-based	No.of other type
DataS	6198	3984	971	1243
DataH	21263	12168	2243	10966
DataK	13000	6267	1142	5585

and Konno *et al.* [48]. This paper refers to the data sets used in these three studies as DataS, DataH, DataK and we compared the performance of our proposed model to other methods over these datasets. Table 1 shows the statistics of these three benchmark datasets.

B. MATERIALS REPRESENTATION

Two most important aspects of any machine learning model are data representation and learning algorithms. There are two main types of methods for material representation: one is based on the molecular formula and the other is based on its crystal structure. The former type requires only chemical composition as input without crystal structure information, which allows to explore the whole chemical composition space. Commonly used material representation methods of this type include Magpie descriptors (with element attribute statistics) and One-hot coding. The structure-based material representation methods refer to the construction of vector-based crystal structure data representations. Commonly used such methods include the use of crystal geometry to construct Vorono-Dirichlet polyhedron (VDP) on each atom [49] and using the radial distribution function (RDF) [50] to characterize the accumulation of atoms in the crystal and the distance between the individual bonds. Another encoding is the graph encoding with CNN [51]. The accuracy of structure-based characterization methods is limited by our ability to perform all the domain knowledge required for feature representation of materials. In this research, we used statistical elemental properties in molecular formulas to characterize materials.

The statistical elemental properties representation refers to the calculation of all statistics of elemental properties of the material, such as the number of cycles of the elements in the numerator on the periodic table, the number of families, the radius of the atom, the melting temperature, average fraction of valence electrons from s, p, d, and f orbitals in all elements. In this paper, we have calculated the 22 kinds of attributes of the elements (See Table 2) using the Matminer package [52].

TABLE 2. 22 Elemental attributes used in materials representation [52].

Atom Number MendeleevNumber AtomicWeight MeltingTemperature Period Table Column Period Table Row	CovalentRadius Electronegativity GSvolume_per atom GSbandgap GSmagmom SpaceGroupNumber
NsValence NpValence NdValence NfValence NValence	NsUnfilled NpUnfilled NdUnfilled NfUnfilled NUnfilled

For each attribute, we calculate its maximum, minimum, range, mean, variance, and mode characteristics of the constituent elements of a given material, so that it can be characterized as a matrix T as shown in (1) at the bottom of this page. $T \in R^{s\times d}(s=22, d=6)$. Each property has six statistical values, which can be regarded as a local representation of the material. Then we use 32 convolution kernels of the size 1×6 to scan the feature matrix T in a row to extract local features.

III. ConvGBDT MODEL

A. CNN BASED FEATURE EXACTION MODEL

CNN is a kind of deep learning algorithm that has led to the breakthrough in computer vision [53] and other applications. Its main advantage is the ability to extract hierarchical features from high-dimensional data. A CNN generally consists of the following six parts: the input layer, convolution layer, activation layer, pooling layer, fully connected layer and output layer. The convolution operator is used to extract local features; pooling operators are used to compress the feature map obtained by convolution to simplify the network and reduce computational complexity; activation function is the main source of nonlinear transformation of the neural network; fully connected layer is used for mapping learned high-level features to the output. The characterization map of the material $T \in R^{s\times d}$ is much smaller than the image feature map. In order to avoid missing features of the pooling operation when extracting main features, CNN in this study does not contain pooling operations. Our CNN structure is shown in Figure 1.

According to formula (1) each row of the material characterization matrix $T \in R^{s\times d}$ is a set of 6 different statistics of a given element property. The CNN model consists of two

$$T = \begin{bmatrix} \text{Numb_min} & \text{Numb_max} & \text{Numb_range} & \cdots & \text{Numb_mode} \\ \text{GSmg_min} & \text{GSmg_max} & \text{GSmg_range} & \cdots & \text{GSmg_mode} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \text{Spac_min} & \text{Spac_max} & \text{Spac_range} & \cdots & \text{Spac_mode} \end{bmatrix} \in R^{22\times 6} \quad (1)$$

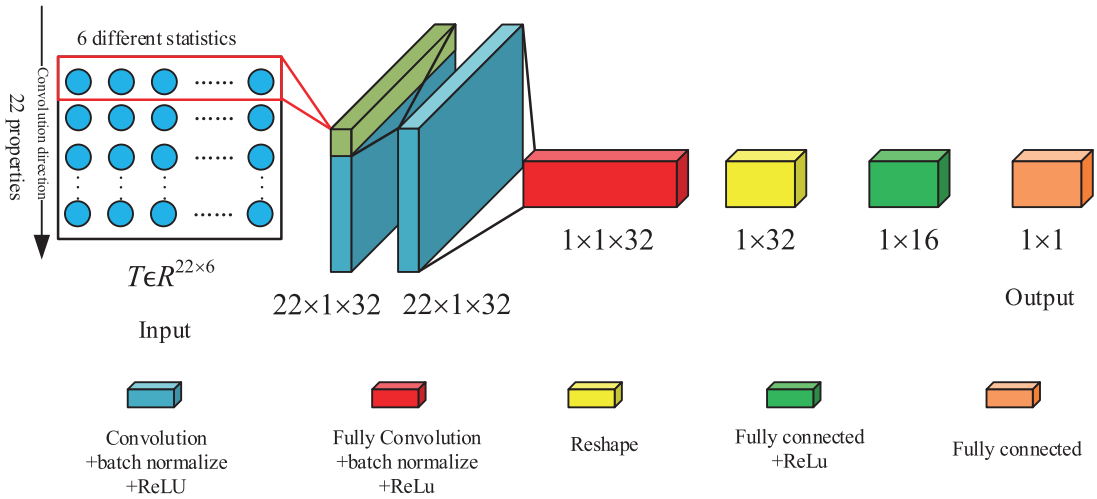


FIGURE 1. CNN model.

TABLE 3. Parameters of CNN model.

Layer	Input Shape	Kernel number	Kernel Size	Stride	Output Shape
Conv1	[batch, 22, 6, 1]	32	(1, 6, 1)	(1, 1)	[batch, 22, 1, 32]
Conv2	[batch, 22, 1, 32]	32	(1, 1, 32)	(1, 1)	[batch, 22, 1, 32]
Conv3	[batch, 22, 1, 32]	32	(22, 1, 32)	(1, 1)	[batch, 1, 1, 32]
Reshape	[batch, 1, 1, 32]	-	-	-	[batch, 32]
Fc4	[batch, 32]	-	-	-	[batch, 16]
Fc5	[batch, 16]	-	-	-	[batch, 1]

row-scanning convolutional layers, one fully convolutional layer and two fully connected layers. The detailed parameters of each layer of the CNN model are shown in Table 3.

The row-scanning convolution kernels of the first convolutional layer are applied to the matrix T to fuse different statistical values for extracting high-level features. The width of the convolution kernel l is d, which is the same as the width of the material characterization matrix T, and the height of the convolution kernel h is set as 1. To extract more relationships among the 22 element properties, we use a fully convolutional layer to learn inter-property features. This idea was originally proposed in [54] for semantic segmentation. The size of the convolution kernel here is the same as the size of the input feature map. The difference between fully convolutional and fully connected layers here is that the feature map of fully convolutional layer is generated by the convolution operation while the fully connected layer is done by weight sums. After each of the convolutional layers, a batch normalization layer [55] is used to improve the convergence speed of the model and reduce the influence of network weight initialization during the learning process. Except for the final output layer, a rectified linear unit (ReLu) is used as the activation function for each layer of the neural network.

B. GRADIENT BOOSTING DECISION TREE (GBDT)

Instead of using the fully-connected layer for final regression, we propose to combine the CNN based feature extraction with a powerful prediction model, the Gradient Boosting Decision Tree (GBDT) for learning the regression model. GBDT is also known as the Multiple Additive Regression Tree (MART), which is a statistical integrated learning method proposed by Friedman [56]. Integrated learning is a commonly used statistical learning method, which learns to combine multiple weak learners effectively to build a strong learner with high prediction accuracy, which can reduce the variance and deviation of the prediction model. The GBDT algorithm can be expressed as a boosting method based on decision trees (DTs):

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J} c_{mj}$$

where $f_m(x)$ represents the m-th learner, c_{mj} represents the loss of the j-th node on the m-th tree. The GBDT model can be trained using the forward distribution algorithm:

(1) Determine the initial decision tree $f_0(x) = arg\ \underset{c}{min} \sum_{i=1}^{N} L(y_i, c)$

(2) For $m = 1, 2, \cdots, M$, where M is the number of trees.

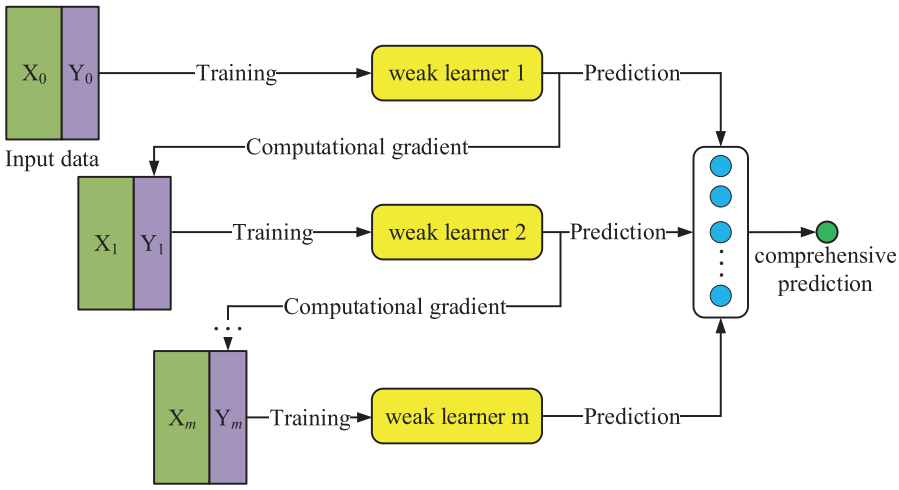


FIGURE 2. The specific process of GBDT.

a) For $i = 1, 2, \cdots, N$, where N is the number of samples, the gradients can be calculated as:

$$r_{mi} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$$

b) Using (x_i, r_{mi}) we can fit a regression tree and get the m-th regression tree. Its corresponding leaf node area is $R_{mj} (j = 1, 2, 3, \cdots, J)$, where J is the number of leaf nodes.

c) For each sample in the leaf node, we find the best output value c_{mj} to minimize the loss function $L(\cdot)$. In the decision tree, the value of leaf nodes has been generated once. The purpose of this step is to slightly change the value of leaf nodes in the decision tree, hoping that the fitting error will become smaller. $c_{mj} = argmin \sum_{x_i \in R_{ij}} L(y_i, f_{m-1}(x_i) + c)$

d) Finally, we can get the decision tree fitting function of this round as: $h_t(x) = \sum_{j=1}^{J} c_{mj}$

(3) The resulting strong learner model can be described as: $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J} c_{mj}$

The overall process of GBDT algorithm is shown in Figure 2.

C. CONVOLUTIONAL GRADIENT BOOSTING DECISION TREE (ConvGBDT)

In this work, we propose a hierarchical feature extraction neural network architecture based on convolutional network layers to extract features from the input data representation, and then use the GBDT to predict the Tc of the superconductors with the features extracted by the convolutional layers. Our method is called Convolutional Gradient Boosting Decision Tree (ConvGBDT). The architecture of our ConvGBDT is shown in Figure 3. First, the convolutional feature extraction network is trained to screen, fuse, and extract features from the input characterization matrix T. The training model is shown in Figure 1, in which the Tc is used as the target values to predict. After training, only the convolution layers and the fully convolutional layers are kept to extract features. And then GBDT is trained to make the final regression predictions based on the output features of the last convolutional layer.

IV. DATA PREPROCESSING AND MODEL TRAINING

A. DATA PREPROCESSING

To reduce the influence of difference scales of the materials attributes on model, we applied the normalization preprocessing step to the characterization matrix. For each materials attribute, we convert the original value into a scaled value as shown in formula (2).

$$x' = \frac{x - min\ A}{max\ A - min\ A} \tag{2}$$

where, x represents original data, x' represents normalized data, max A and min A represents the maximum and minimum value of the attribute.

B. MODEL TRAINING AND COMPUTIONAL EXPERIMENTAL ENVIROMENTS

To evaluate the performance of the regression models, we use the mean absolute error (MAE), the root mean square error ($RMSE$), and R-Squared (R^2) as the evaluation measures. These performance measures can be calculated as follows:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|y_i - \hat{y}_i\right| \tag{3}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{m}\left(y_i - \bar{y}\right)^2} \tag{5}$$

where, m is the number of samples, y_i and $\hat{y}_i$ are the true and predicted values of the i sample label (the Tc of the

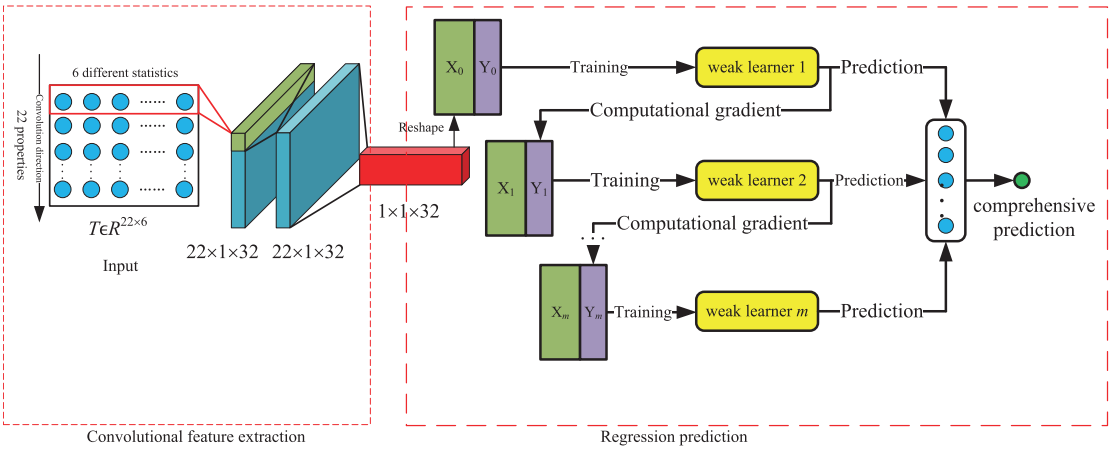


FIGURE 3. Architecture of ConvGBDT algorithm.

superconducting material), $\bar{y}$ is the average of the m sample real labels.

To train the CNN and ConvGBDT models, the data set needs to be divided into a training set, a validation set and a test set. The training set is used to update the weights of the models, the validation set is used to adjust the hyper-parameters of the models, and the test set is used to judge whether a model is good or not. Figure 4 shows the data partitioning process using DataS as an example. We first applied the 10-fold cross-validation to the DataS, where in each fold the dataset is split into a training set "A" and a test set (10% of all samples). In each fold, we further randomly divide dataset "A" into a training set "a" with 90% samples and a validation set with 10% samples. Training set "a" and validation set are used to train and tune the parameters of the CNN and ConvGBDT models, and the test set is used to evaluate the quality of these two models, as shown in Figure 4. When training CNN, we keep the best performing model within 700 epochs in terms of R^2 over the validation set as the final model, which is then combined with GBDT to obtain the final ConvGBDT model.

Since the decision tree is a non-differentiable model, the ConvGBDT model cannot be directly trained by the gradient descent method. Thus, we first train the CNN model using Tc as the prediction target (See Figure 1), and then train the ConvGBDT model by using the output features from the CNN module as input and the Tc values as target values. Here the neural network hyper-parameters mainly include momentum, learning rate, optimization algorithms and the batch size. The hyper-parameter of GBDT mainly includes the number of trees, learning rate, sampling rate, and max depth of the decision trees. Among them, the learning rate is one of the most important hyper-parameters of deep neural networks, and we have tried the learning rate from 0.1 to $1e^{-6}$(each time reduced by 10 times).

In order to ensure the stability and reliability of the computational experimental results, ConvGBDT and all subsequent comparative computational experiments (RF, GBDT, etc.) were subjected to 10 times 10-fold cross-validation to

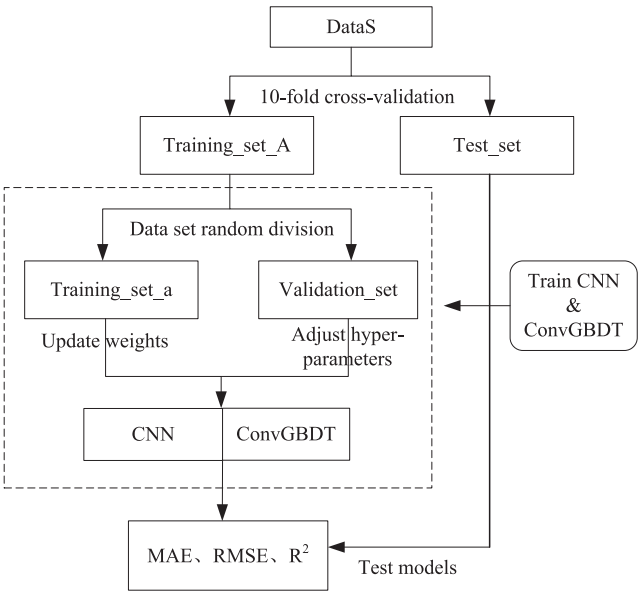


FIGURE 4. The process of training ConvGBDT model and evaluating its performance. In each fold of cross-validation, the training set A is split into a subset a with 90% samples and validation_set with 10% samples. Both datasets are used to train and tune the CNN model with the best CNN model is picked to be combined with GBDT to be further tuned with the validation_set to build the final ConvGBDT.

calculate the average performances. The whole model is developed based on Python 2.7. The CNN model uses the Tensorflow9.0 [57] deep learning framework. The implementation of the baseline machine learning algorithms is based on Scikit-learn [58]. All the programs except the baseline machine learning algorithms are run on a Dell Server with 3.6GHz GPU and NVIDIA GPU GTX1080Ti.

V. COMPUTATIONAL EXPERIMENTAL RESULTS

A. HYPERPARAMETER OF THE MODEL

We set the initial values for each hyper-parameter based on empirical intuition and then used a greedy algorithm to adjust each hyper-parameter step by step instead of performing a

TABLE 4. Hyper-parameters of ConvGBDT model.

Batch size	CNN_Lr	Tree number
32	0.01	400
GBDT_Lr	Max depth	Sampling Rate
0.06	20	0.4

grid search, which is not feasible due to computational cost. We adjust the hyperparameters based on the loss values on the test set to obtain a set of hyperparameters that minimize the loss values on the test set. Finally, all the hyper-parameters of the ConvGBDT model are determined as shown in Table 4:

B. COMPARISON AND ANSLYSIS

We compare the proposed ConvGBDT with three regression models including CNN, GBDT and RF. The results are shown in Table 5. The row 3 is the result of GBDT with the 132-dimensional vectors as input flattened from the representation matrix T. The rows 1, 6, 7, 8 are the results of the corresponding models with the element statistical matrix T as input, row 2 is the result of Konno *et al.* [48], row 4 is the result of Hamidieh [33], and row5 is the result of Stanev *et al.* [31]. Figure 5 shows the Training and validation of R^2, RMSE and MAE change during training CNN.

First, from row 1, we can find that the CNN model achieves an RMSE of 7.889 on the test dataset, which is lower than the results in [29] on the same dataset obtained by an ensemble Random Forest algorithm. Their algorithm combines the models built on Magpie descriptors (large sampling, but features limited to compositional data) and AFLOW features (small sampling, but diverse and pertinent features). We also found that compared to RMSE of 5.989 on the training set, the test RMSE (7.889) is much worse and cannot be further reduced despite of our extensive efforts of parameter and architecture tuning of the CNN model. Row 3 of Table 5 shows the performance of GBDT algorithm over DataS, the original characterization data (features) of the superconductors. Its RMSE is 6.92, better than the CNN method. Both the standard CNN and GBDT models did not get good results on the test set. Row 6 shows the performance of our ConvGBDT model with an RMSE of 5.512, which is much better than the results of either CNN or GBDT or the result in [29]. It actually obtained the best results in terms of all three criteria: RMSE, MAE, R^2. This indicates that the CNN feature extraction part of the model can learn more effective features from the material's characterization matrix T through the layers of convolutional integration, filtering, and compression, which allows the GBDT model to more effectively model the relationship between the features and Tc. The reason why ConvGBDT is better than RF and GBDT is because we have designed a hierarchical representation matrix T of superconductors (each row represents a different attribute statistics such as average fraction of valence electrons from s, p, d, and f orbitals in all elements.), we perform feature extraction on the matrix T by row-scanning convolution kernels, and

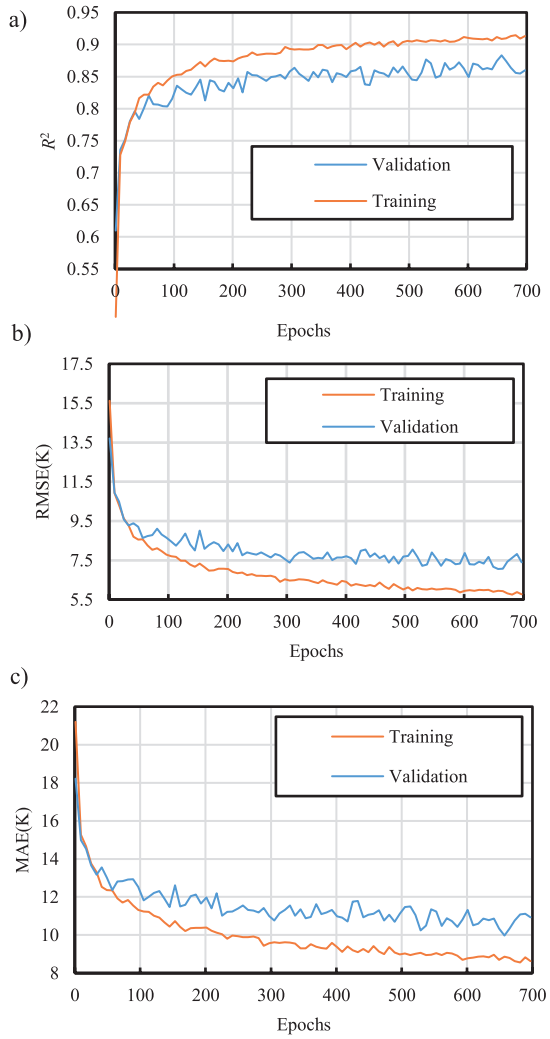


FIGURE 5. Training and validation errors during training.

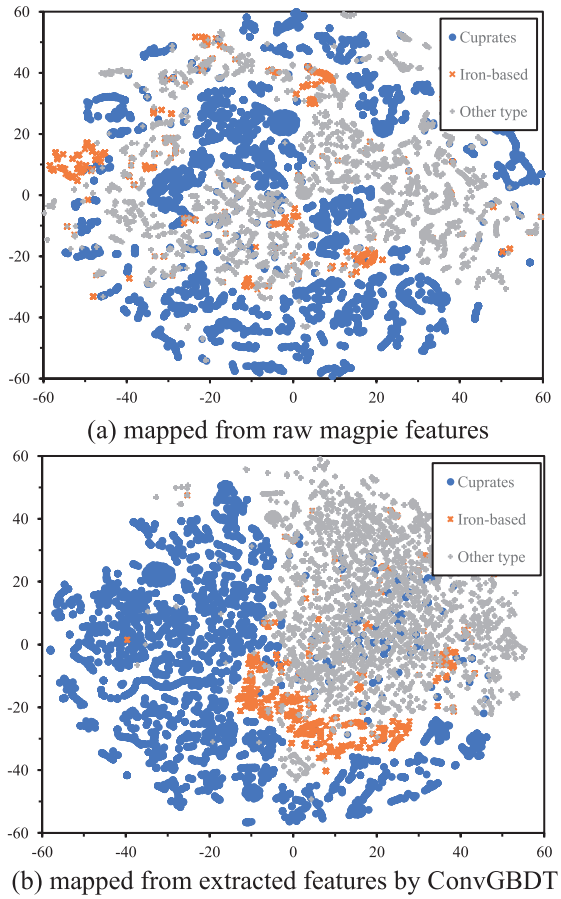
then fuse the features using a fully convolutional layer, so that we can find the similarity between the material composition similarity and Tc. The relationship can be seen from Figure 6. However, non-parametric machine learning methods such as RF and GBDT do not have the ability to extract features. The prediction tree is directly established on the original features, which places high requirements on the algorithm. Therefore, RF/GBDT is naturally not as good as ConvGBDT, which uses CNN for feature extraction and then inputs GBDT for regression prediction. Row 5, 6 indicate that the results of our ConvGBDT method on the DataS dataset are better than those of Stanev *et al.* [31] for which our ConvGBDT achieves R^2 of 0.907 compared to 0.876 of Stanev *et al.* [31]. The results in row 4, 7 show that the results of the ConvGBDT on DataH dataset are better than those of Hamidieh [33]. Similarly, the results in row 2, 8 demonstrate the higher performance of ConvGBDT over DataK data set compared to those of Konno *et al.* [48]. Overall, our ConvGBDT method has achieved the best results on all three public datasets: DataS, DataH, DataK, indicating the success of merging strategy of CNN with GBDT for Tc prediction of superconductors.

TABLE 5. Comparison of model performances in terms of RMSE, MAE, R^2.

Number	Model/Ref	Dataset	RMSE_tr(K)	MAE_tr(K)	R^2_tr	RMSE(K)	MAE(K)	R^2
1	CNN	DataS	5.989	8.878	0.908	7.889	12.021	0.831
2	[48]	DataK	-	-	-	-	-	0.930
3	GBDT	DataS	-	-	-	6.920	10.926	0.873
4	[33]	DataH	-	-	-	9.500	-	0.920
5	[31]	DataS	-	-	-	-	-	0.876
6	ConvGBDT	DataS	0.127	0.376	0.998	**5.512**	**8.930**	**0.907**
7	ConvGBDT	DataH	1.805	4.260	0.985	**4.653**	**8.695**	**0.937**
8	ConvGBDT	DataK	1.838	4.320	0.984	**4.746**	**8.834**	**0.931**

TABLE 6. List of symbols and acronyms.

Symbols/ Acronyms	Descriptions
Tc	critical temperature.
$T \in R^{s \times d}$	characterization matrix of superconductors.
$f_m(x)$	m-th learner.
s	The number of attributes of the elements
d	The number of statistical values
J	the number of leaf nodes
M	the number of trees.
c_{mj}	The loss of the j-th node on the m-th tree
$L(\cdot)$	Loss function
MAE	mean absolute error
$RMSE$	root mean square error
R^2	R-Squared-
CNN	convolutional neural network
GBDT	gradient boosting decision tree
ConvGBDT	convolutional gradient boosting decision tree
DFT	density functional theory
BSDFT	beyond standard density functional theory
MP	Materials Project
OQMD	Open Quantum Materials Database
AFLOWLIB	Automatic Flow of Materials Discovery Library
NoMaD	Novel Materials Discovery Database
RF	random forest
VDP	Vorono-Dirichlet polyhedron
RDF	radial distribution function
ReLu	rectified linear unit
MART	Multiple Additive Regression Tree
DT	decision tree
BSDFT	beyond standard density functional theory
SC	Superconducting material



(a) mapped from raw magpie features

(b) mapped from extracted features by ConvGBDT

FIGURE 6. Comparison of distributions of superconductor materials of different categories with (a) raw magpie features (b) features extracted by CNN. The axis X1 and X2 are the t-sne mapped 2D dimensions without explicit simple physical meaning.

Superconducting materials (SCs) can be roughly classified into cuprates-based, iron-based, and all other unconventional superconductors. A large amount of research has been focused on cuprates and iron-based compounds. To illustrate that the convolutional layer of our ConvGBDT model obtains useful features from the characterization matrix T, we randomly extract 50% of the samples from DataH to visualize the original input matrix T and the eigenvectors extracted by the convolutional layer into 2 dimension space using T-sne [59], a tool for visualizing high-dimension data. Compared to Principal Components Analysis, T-sne preserves only small pairwise distances or local similarities in its manifold learning embedding process so that it has better capability to make the relative distances among data samples in the lower dimension to be more consistent to their distances in the high dimension. As can be seen from Fig. 6(a), different categories of superconductor materials

are mixed together when the raw elemental features of the input matrix representations are fed to T-sne for dimension reduction.

However, when we map the superconductor materials into 2D space using the extracted features by the convolutional layers, these materials can be roughly divided into three regions (Fig. 6(b)) with much better separations among categories. This also explains the higher prediction performance of our ConvGBDT algorithm. The key reason is that the feature space are composed of mixed types of superconductors, whose Tc values tend to be very different from each other. So when the RF tries to make predictions based on its neighbors of the query sample, it will be misled to give erroneous *Tc* predictions. On the other hand, in our ConvGBDT, the learned representation allows the different types of superconductors to cluster together, and when the GBDT is used to make predictions, it tends to use the neighbor samples of the query materials which belong to the same category and thus have much similar Tc values, leading to better performance.

C. DISCUSSION

Because deep learning has stronger generalization ability than machine learning models, we can use our proposed deep learning model to predict the Tc without using the DFT, and find that new superconductors. The first step in discovering new materials using deep learning methods is to establish an accurate material attribute prediction model, then construct an imaginary material space(Such as AxByCz x+y+z $<$ 10, where A, B and C are different elements, x, y and z are the subscripts of the corresponding elements), finally build an accurate prediction model to screen for possible new materials on this space. For example, After using FNN to build an accurate prediction model of formation energy, Jha *et al.* [60] screens out materials with low formation energy in the constructed material paradigm. After establishing a *Tc* prediction model using RF, Stanev *et al.* [31] Screened on ICSD database to find possible superconductors. Therefore, the model for predicting *Tc* proposed in this paper can be used to discover new superconductors.

When the neural network makes predictions, it is necessary to ensure that the distribution of the predicted data is consistent with the distribution of the training data. In the process of discovering new materials, it is often to use an exhaustive method to establish a hypothetical material space. This does not guarantee the consistency of the distribution of prediction data and training data. Generative Adversarial Network (GAN) is a model for learning data distribution. We can use it to generate hypothetical materials consistent with the distribution of training materials. Then use our prediction model to screen the generated hypothetical material to find promising new materials.

VI. CONCLUSION

In this paper, we propose ConvGBDT, a novel deep learning algorithm for *Tc* prediction. It combines the advantage of CNN for hierarchical representation/feature learning and the modeling power of GBDT decision tree models. The materials are encoded based on their elemental properties from their composition formulas. By replacing the fully connected layer of standard CNN with the GBDT decision trees, we developed the ConvGBDT prediction model of *Tc*. Extensive computational experiments showed that our ConvGBDT model achieved the best results on three superconductor data sets including DataS, DataH, and DataK. Distribution visualization of the superconductors of different categories using the raw elemental features and the CNN-extracted features shows that ConvGBDT can learn more effective features that can distinguish between cuprate and iron-based superconductors much better than the raw features. The higher *Tc* prediction performance of ConvGBDT also shows that the GBDT decision tree can better capture the mapping relationship between the features extracted by the convolutional layers and the *Tc*. All the source code of the ConvGBDT model and related datasets are publicly accessible at http://www.mekhub.cn/danyabo/superconductor.

REFERENCES

[1] A. Gurevich, "To use or not to use cool superconductors?" *Nature Mater.*, vol. 10, no. 4, pp. 255–259, Apr. 2011.

[2] K. Yasuda, A. Ichinose, A. Kimura, K. Inoue, H. Morii, Y. Tokunaga, S. Torii, T. Yazawa, S. Hahakura, K. Shimohata, and H. Kubota, "Research & development of superconducting fault current limiter in Japan," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 2, pp. 1978–1981, Jun. 005.

[3] Y.-S. Jo, K.-S. Ryu, and M. Park, "1st phase results and future plan of DAPAS program," *IEEE Trans. Appl. Supercond.*, vol. 16, no. 2, pp. 678–682, Jun. 2006.

[4] T. Verhaege, P. F. Herrmann, J. Bock, L. Cowey, G. Moulaert, H. C. Freyhardt, A. Usoskin, J. Paasi, and M. Collet, "European project on a self-limiting superconducting power link," *Superconductor Sci. Technol.*, vol. 13, no. 5, pp. 488–492, May 2000.

[5] M. Granovetter and P. McGuire, "The making of an industry: Electricity in the United States," *Sociol. Rev.*, vol. 46, no. 1, pp. 147–173, May 1998.

[6] J.-W. Yeh, S.-K. Chen, S.-J. Lin, J.-Y. Gan, T.-S. Chin, T.-T. Shun, C.-H. Tsau, and S.-Y. Chang, "Nanostructured high-entropy alloys with multiple principal elements: Novel alloy design concepts and outcomes," *Adv. Eng. Mater.*, vol. 6, no. 5, pp. 299–303, May 2004.

[7] A. Sharma, P. Singh, D. D. Johnson, P. K. Liaw, and G. Balasubramanian, "Atomistic clustering-ordering and high-strain deformation of an $Al_{0.1}CrCoFeNi$ high-entropy alloy," *Sci. Rep.*, vol. 6, Aug. 2016, Art. no. 31028.

[8] A. Sharma, S. A. Deshmukh, P. K. Liaw, and G. Balasubramanian, "Crystallization kinetics in AlxCrCoFeNi ($0 \leq x \leq 40$) high-entropy alloys," *Scripta Mater.*, vol. 141, pp. 54–57, Dec. 2017.

[9] A. Sharma and G. Balasubramanian, "Dislocation dynamics in $Al_{0.1}CoCrFeNi$ high-entropy alloy under tensile loading," *Intermetallics*, vol. 91, pp. 31–34, Dec. 2017.

[10] S. Fritzsche, "Relativistic many-body theory: A new field-theoretical approach," *Int. J. Quantum Chem.*, vol. 112, no. 14, pp. 2688–2689, Jul. 2012.

[11] M. van Schilfgaarde, T. Kotani, and S. Faleev, "Quasiparticle self-consistent *GW* theory," *Phys. Rev. Lett.*, vol. 96, no. 22, 2006, Art. no. 226402.

[12] C. W. Castleton, A. Höglund, and S. Mirbt, "Managing the supercell approximation for charged defects in semiconductors: Finite-size scaling, charge correction factors, the band-gap problem, and the *ab initio* dielectric constant," *Phys. Rev. B, Condens. Matter*, vol. 73, no. 3, 2006, Art. no. 035215.

[13] H. Koinuma and I. Takeuchi, "Combinatorial solid-state chemistry of inorganic materials," *Nature Mater.*, vol. 3, no. 7, pp. 429–438, Jul. 2004.

[14] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "Commentary: The materials project: A materials genome approach to accelerating materials innovation," *APL Mater.*, vol. 1, no. 1, Jul. 2013, Art. no. 011002.

[15] A. Belsky, M. Hellenbrandt, V. L. Karen, and P. Luksch, "New developments in the inorganic crystal structure database (ICSD): Accessibility in support of materials research and design," *Acta Crystallographica B, Struct. Sci.*, vol. 58, no. 3, pp. 364–369, Jun. 2002.

[16] J. Hafner, C. Wolverton, and G. Ceder, "Toward computational materials design: The impact of density functional theory on materials research," *MRS Bull.*, vol. 31, no. 9, pp. 659–668, Sep. 2006.

[17] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, "Materials design and discovery with high-throughput density functional theory: The open quantum materials database (OQMD)," *JOM*, vol. 65, no. 11, pp. 1501–1509, Nov. 2013.

[18] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, "The open quantum materials database (OQMD): Assessing the accuracy of DFT formation energies," *npj Comput. Mater.*, vol. 1, no. 1, Dec. 2015, Art. no. 15010.

[19] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. H. Taylor, L. J. Nelson, G. L. W. Hart, S. Sanvito, M. Buongiorno-Nardelli, N. Mingo, and O. Levy, "AFLOWLIB. ORG: A distributed materials properties repository from high-throughput *ab initio* calculations," *Comput. Mater. Sci.*, vol. 58, pp. 227–235, Jun. 2012.

[20] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.

[21] Z. Yu, F. Liu, R. Liao, Y. Wang, H. Feng, and X. Zhu, "Improvement of face recognition algorithm based on neural network," in *Proc. 10th Int. Conf. Measuring Technol. Mechtron. Automat. (ICMTMA)*, Feb. 2018, pp. 229–234.

[22] A. Agrawal and A. Choudhary, "Perspective: Materials informatics and big data: Realization of the 'fourth paradigm' of science in materials science," *APL Mater.*, vol. 4, no. 5, p. 053208, 2016.

[23] A. J. Hey, S. Tansley, and K. M. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Redmond, WA, USA: Microsoft Research Redmond, 2009.

[24] K. Rajan, "Materials informatics: The materials 'gene' and big data," *Annu. Rev. Mater. Res.*, vol. 45, pp. 153–169, Jul. 2015.

[25] J. Hill, G. Mulholland, K. Persson, R. Seshadri, C. Wolverton, and B. Meredig, "Materials science with large-scale data and informatics: Unlocking new opportunities," *MRS Bull.*, vol. 41, no. 5, pp. 399–409, May 2016.

[26] L. Ward and C. Wolverton, "Atomistic calculations and materials informatics: A review," *Current Opinion Solid State Mater. Sci.*, vol. 21, no. 3, pp. 167–176, Jun. 2017.

[27] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, "Machine learning in materials informatics: Recent applications and prospects," *npj Comput. Mater.*, vol. 3, no. 1, p. 54, Dec. 2017.

[28] Z. D. Pozun, K. Hansen, D. Sheppard, M. Rupp, K.-R. Müller, and G. Henkelman, "Optimizing transition states via kernel-based machine learning," *J. Chem. Phys.*, vol. 136, no. 17, May 2012, Art. no. 174101.

[29] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Machine learning of molecular electronic properties in chemical compound space," *New J. Phys.*, vol. 15, no. 9, 2013, Art. no. 095003.

[30] A. Agrawal, P. D. Deshpande, A. Cecen, G. P. Basavarsu, A. N. Choudhary, and S. R. Kalidindi, "Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters," *Integrating Mater. Manuf. Innov.*, vol. 3, no. 1, pp. 90–108, Dec. 2014.

[31] V. Stanev, C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, and I. Takeuchi, "Machine learning modeling of superconducting critical temperature," *npj Comput. Mater.*, vol. 4, no. 1, p. 29, Dec. 2018.

[32] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Comput. Mater.*, vol. 2, no. 1, p. 16028, Nov. 2016.

[33] K. Hamidieh, "A data-driven statistical model for predicting the critical temperature of a superconductor," *Comput. Mater. Sci.*, vol. 154, pp. 346–354, Nov. 2018.

[34] Z. Zhang, Y. Xu, L. Shao, and J. Yang, "Discriminative block-diagonal representation learning for image recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3111–3125, Jul. 2018.

[35] Q. Zhao, S. Lyu, B. Zhang, and W. Feng, "Multiactivation pooling method in convolutional neural networks for image recognition," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–15, Jun. 2018.

[36] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.

[37] D. E. Reich, "Method and system for identifying and correcting accent-induced speech recognition difficulties,"U.S. Patent 8 285 546 B2, Oct. 9, 2012.

[38] P. Hastings, M. A. Britt, K. Rupp, K. Kopp, and S. Hughes, "Deep and shallow natural language understanding for identifying explanation structure," in *Deep Learning: Multi-Disciplinary Approaches*. Abingdon, U.K.: Taylor & Francis, 2019.

[39] J. Ganitkevitch, "Large-scale paraphrasing for natural language understanding," in *Proc. NAACL HLT Student Res. Workshop*, 2013, pp. 62–68.

[40] S. Li, G. Liu, X. Tang, J. Lu, and J. Hu, "An ensemble deep convolutional neural network model with improved D-S evidence fusion for bearing fault diagnosis," *Sensors*, vol. 17, no. 8, p. 1729, 2017.

[41] Y. Yao, H. Wang, S. Li, Z. Liu, G. Gui, Y. Dan, and J. Hu, "End-to-end convolutional neural network model for gear fault diagnosis based on sound signals," *Appl. Sci.*, vol. 8, no. 9, p. 1584, 2018.

[42] A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi, and L. Song, "Material structure-property linkages using three-dimensional convolutional neural networks," *Acta Mater.*, vol. 146, pp. 76–84, Mar. 2018.

[43] R. Kondo, S. Yamakawa, Y. Masuoka, S. Tajima, and R. Asahi, "Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics," *Acta Mater.*, vol. 141, pp. 29–38, Dec. 2017.

[44] J. Ling, M. Hutchinson, E. Antono, B. DeCost, E. A. Holm, and B. Meredig, "Building data-driven models with microstructural images: Generalization and interpretability," *Mater. Discovery*, vol. 10, pp. 19–28, Dec. 2017.

[45] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018.

[46] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet—A deep learning architecture for molecules and materials," *J. Chem. Phys.*, vol. 148, no. 24, 2018, Art. no. 241722.

[47] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Commun.*, vol. 8, no. 1, p. 13890, Apr. 2017.

[48] T. Konno, H. Kurokawa, F. Nabeshima, Y. Sakishita, R. Ogawa, I. Hosako, and A. Maeda, "Deep learning model for finding new superconductors," 2018, *arXiv:1812.01995*. [Online]. Available: http://arxiv.org/abs/1812.01995

[49] V. A. Blatov, "Voronoi–Dirichlet polyhedra in crystal chemistry: Theory and applications," *Crystallogr. Rev.*, vol. 10, no. 4, pp. 249–318, Oct. 2004.

[50] S. Honrao, B. E. Anthonio, R. Ramanathan, J. J. Gabriel, and R. G. Hennig, "Machine learning of ab-initio energy landscapes for crystal structure predictions," *Comput. Mater. Sci.*, vol. 158, pp. 414–419, Feb. 2019.

[51] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Phys. Rev. Lett.*, vol. 120, no. 14, Apr. 2018, Art. no. 145301.

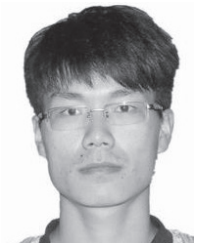
[52] L. Ward, A. Dunn, A. Faghaninia, N. E. R. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K. A. Persson, G. J. Snyder, I. Foster, and A. Jain, "Matminer: An open source toolkit for materials data mining," *Comput. Mater. Sci.*, vol. 152, pp. 60–69, Sep. 2018.

[53] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time-series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1995.

[54] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[56] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, Oct. 2001.
[57] S. S. Girija, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," Tech. Rep., 2016.
[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[59] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[60] D. Jha, L. Ward, A. Paul, W.-K. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, "ElemNet: Deep learning the chemistry of materials from only elemental composition," *Sci. Rep.*, vol. 8, no. 1, Dec. 2018, Art. no. 17593.



YABO DAN received the B.S. degree in mechanical engineering from the Hubei University of Arts and Science, Xiangyang, China, in 2017. He is currently pursuing the M.S. degree in mechanical engineering with Guizhou University, Guiyang, China.

Since 2017, he has been a Research Assistant with the Materials Informatics Group, Guizhou University. His research interests include deep learning, machine learning, and materials informatics.

RONGZHI DONG received the B.S. degree in mechanical engineering from China Agricultural University, Beijing, in 2017. She is currently pursuing the M.S. degree in mechanical engineering with Guizhou University, Guiyang, China.

Since 2018, she has been a Research Assistant with the Materials Informatics Group, Guizhou University. Her research interests include deep learning, machine learning, and materials informatics.

ZHUO CAO received the B.S. degree in mechanical engineering from Xuchang University, Xuchang, China, in 2016. He is currently pursuing the M.S. degree in mechanical engineering with Guizhou University, Guiyang, China. His research interests include deep learning, fault diagnosis, and materials informatics.

XIANG LI received the B.S. degree in mechatronic engineering from Changzhou University, Changzhou, China, in 2017. He is currently pursuing the M.S. degree in mechanical engineering with Guizhou University, Guiyang, China. His research interests include machine learning, deep learning, material informatics, and perovskite materials discovery.

CHENGCHENG NIU received the B.S. degree in mechanical engineering from Shandong Jiaotong University, Jinan, China, in 2017. She is currently pursuing the M.S. degree in mechanical engineering with Guizhou University, Guiyang, China.

Since 2017, she has been a Research Assistant with the Materials Informatics Group, Guizhou University. Her research interests include machine learning, deep learning, neural networks, and materials informatics.

SHAOBO LI received the Ph.D. degree in computer software and theory from the Chinese Academy of Sciences, China, in 2003. From 2007 to 2015, he was the Vice Director of the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, GZU. He has been the Dean of the School of Mechanical Engineering, GZU, since 2015. He is currently a Professor with the School of Mechanical Engineering, Guizhou University (GZU), China. He is also a part time Doctoral Tutor with the Chinese Academy of Sciences. He has published more than 200 articles in major journals and international conferences. His current research interests include big data of manufacturing and intelligent manufacturing. His research has been supported by the National Science Foundation of China and the National High-Tech Research and Development Program (863 Program). He received honors and awards from New Century Excellent Talents in the University of Ministry of Education of China, an excellent expert and innovative talent of Guizhou, the Group Leader of manufacturing information, and an alliance Vice Chairman of intelligent manufacturing industry in Guizhou.

JIANJUN HU received the B.S. and M.S. degrees in mechanical engineering from the Wuhan University of Technology, China, in 1995 and 1998, respectively, and the Ph.D. degree in computer science from Michigan State University, in 2004, in the area of machine learning and evolutionary computation. He worked as Postdoctoral Fellow at Purdue University and the University of Southern California, from 2004 to 2007. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of South Carolina at Columbia, Columbia, SC, USA. His research interests include machine learning, deep learning, data mining, evolutionary computation, fault diagnosis, bioinformatics, and material informatics. He is also an Associate Editor of *Nature*, *Scientific Report*, *PLOS One*, and *BMC Bioinformatics*.