Multi-Channel Graph Convolutional Network based End-Point Element Composition Prediction of Converter Steelmaking

S. C. Chang*, C. H. Zhao*, Y. L. Li**, M. Zhou**, C. Fu**, H. L. Qiao**

** State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China (email: chhzhao@zju.edu.cn)*
*** Alibaba Group, Hangzhou, 310024, China*

Abstract: End-point composition is an important quality standard for the converter steelmaking process, which consists of multiple elements, including C, Si, Mn, etc. However, it is hard to measure the element composition online. Real-time and precise prediction for element composition is essential for the optimization of alloy addition so as to bring economic profits. Nevertheless, most conventional models neglect the correlations among element compositions and predict each element composition without the information from other elements. In this paper, a new multi-channel graph convolutional network is proposed to integrate these correlations with the process variables together for a more accurate prediction model. The proposed model uses graph structure to describe the correlations among element compositions. Specifically, through the multi-channel design, each element composition can be learned based on process variables in an independent channel. Element compositions and correlations among them are respectively described by nodes and edges in graph. With the constructed graph, the graph convolution across channels can fuse the features of correlated elements to explicitly exploit the correlation information for performance improvement. Besides, compared with conventional methods which learn relations among nodes based on distances, we take sparse representation learned by sparse coding as edges to describe the correlations among nodes. As strong correlations exist among element compositions, the consideration of correlation information can integrate the learning of correlated elements and bring performance improvement. Experiments based on the real converter steelmaking process demonstrate the superiority and effectiveness of the proposed model.

Keywords: converter steelmaking, end-point composition prediction, graph convolutional network.

1. INTRODUCTION

Converter steelmaking is an essential production process in industry. Liquid iron, scrap steel and iron alloy are the main production materials and they chemically react in the converter without heating (Yu et al., 2001). Meanwhile, oxygen lance needs to blow oxygen into the converter to regulate the reaction. When the reaction reaches the end-point, the converter rotates and pours out the liquid steel into steel ladles. Element compositions at end-point are main quality standards closely related to the quality of steel. When the element compositions in steel ladles deviate from standards, specific alloys are added into steel ladles to adjust element compositions at end-point. With the precise predictions for the end-point element compositions, the amount of alloy addition can be optimized rather than be decided manually. Thus, huge economic benefits can be made from the cost saving of alloy addition for steel factories.

In recent years, many end-point element composition prediction methods have been proposed. They can be categorized into three classes, namely, the first-principle method, the data-driven method and the mixed method of the first two (Yan et al., 2016). Due to the complexity of converter steelmaking process, model accuracy and huge computation burden are limitations of the first-principle method. Compared with the first-principle method, data-driven method only requires little mechanism knowledge about complex process and develops rapidly. In general, various data can be used to construct data-driven prediction model, including indexes of production materials, process variables, manipulated variables and flame images. Classic data-driven methods like principal component regression (PCR), partial least squares (PLS), support vector regression (SVR) and artificial neural network (ANN) have been applied to predict the end-point element composition of converter steelmaking. He et al. (2018) used the dimension-reduction property of principal component analysis (PCA) to extract latent variables first. Then back propagation neural network (BP) was constructed to predict phosphorus content. But this two-step pattern neglects the correlation between latent variables and predicted variables. To take the correlation into consideration while extracting latent variables, Liu et al. (2007) adopted PLS to analyse the influence factors on phosphorus content and established prediction model based on PLS to prove its effectiveness. As a powerful machine learning model, SVR can easily implement nonlinear fitting by kernel trick. Xu et al. (2011) applied SVR to predict carbon content based on furnace flame images. With massive learnable parameters, ANN is more suitable for complex process than other prediction models. Li et al. (2011)

proposed a BP neural network based on Levenberg-Marquardt (LM) algorithm to predict phosphorus content and the hitting rate proved to be high. Due to the strong fitting ability of radial basis function (RBF) neural network, Dong et al. (2014) constructed a RBF neural network to learn the nonlinear and complex relationship between process variables and end-point carbon content. In addition, Han et al. (2014) proposed an evolving membrane algorithm optimized extreme learning machine (ELM) to predict end-point carbon content, which can avoid overfitting and find the global optimal model.

These aforementioned methods mainly focus on how to enhance the fitting ability from process variables to element compositions by different models and optimization techniques while neglecting the correlations among element compositions. Generally, multiple elements are involved in the same reaction, so the variations of their compositions will be highly correlated and coupled. Thus, correlations among element compositions can describe the process property and it is reasonable to predict each element composition with the information of correlated elements. With the correlation information taken into consideration, the cooperative learning among element compositions can be conducted to improve the end-point element composition prediction.

As for the consideration of correlation information, graph convolutional network (GCN) is a machine learning model designed for mining relation described by graph. The relations among nodes are depicted by edges in graph, and graph convolution can learn feature representation based on this relation information. Graph convolution defines the weighted average of neighbouring nodes rather than pixels in receptive field as the operation result. Thus, convolutional neural network (CNN (Lecun et al., 1989)) can be seen as a special case of GCN in Euclidean space. According to the form of graph convolution, GCN can be categorized into two classes, spectral-based and spatial-based (Wu et al., 2020). Bruna et al. (2013) first proposed graph convolution based on spectral graph theory. But the computational complexity of eigen-decomposition problem limits the application of spectral-based GCN on large graphs. Thus, Defferrard et al. (2016) proposed ChebNet which made several approximations and simplifications to alleviate the complex computation. Then, Kipf et al. (2016) introduced a first-order approximation of ChebNet to improve computational efficiency further. The concise computing formula can implement semi-supervised node learning and can be considered as aggregating feature information from neighbouring nodes in a spatial-based perspective. Due to high computing efficiency and interpretability, it has been widely applied as a popular GCN model in social relationship analysis, document classification, etc. Compared with spectral-based GCN, spatial-based GCN essentially propagates node information along edges to extract representation of a graph. Micheli (2009) first proposed a spatial-based GCN named neural network for graphs (NN4G) which also applied residual connections and skip connections to memorize information over each layer. But the unnormalized adjacency matrix used in NN4G may cause hidden node states in extremely different scales. Atwood et al. (2016) proposed a diffusion convolutional neural network (DCNN) assuming that the information diffuse among nodes with a certain transition probability. Because DCNN is designed to capture local behaviour in graph, long-range spatial dependencies or other nonlocal graph behaviour may not be considered.

To the authors' knowledge, GCN model has not been developed for industrial quality prediction application. Besides, these models mentioned above generally take samples as nodes to mine the relations among samples instead of output variables. Thus, each output variable is modelled in the same way without the consideration of correlations among them. In addition, the relations among nodes are generally defined based on distances among nodes which fail to describe correlations among variables. For the prediction task of end-point element composition, correlations among element compositions are helpful for performance improvement and the conventional GCN models cannot be directly applied to integrate these correlations into prediction model. Therefore, new network structure needs to be designed to take the correlations among element compositions into consideration.

In this paper, a new multi-channel GCN model is proposed to integrate correlations among element compositions for the end-point prediction task of converter steelmaking. The relation-mining property of GCN is applied to represent the correlations among element compositions through graph structure. Specifically, in multi-channel network structure, each element composition can be learned based on process variables independently in each channel. Element compositions are also taken as nodes in graph where edges represent the correlations among them. With the constructed graph, correlated channels are connected and share their features through graph convolution across channels. Besides, we adopt sparse coding to learn sparse representation among element compositions, so edges will be able to describe the correlations among nodes. Thus, through sparse coding for graph structure learning and the design of multi-channel network, the proposed model integrates the correlations among element compositions with the process variables together for overall performance improvement.

The contributions of this paper are summarized as below:

- Based on a new multi-channel structure design, GCN is first introduced into the industrial application for the end-point composition prediction task of converter steelmaking process, which exploits the correlations among element compositions for performance improvement.
- The proposed method proposes sparse coding instead of distance index to learn element composition graph so that the correlations among element compositions can be well represented by graph structure.

2. RELATED WORK

The most classic GCN model is proposed by Kipf et al. (2016). A graph consists of multiple nodes and edges.

Generally, samples are taken as nodes and each node has multiple features. Thus, the features of all the nodes can be denoted by feature matrix, $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. m and n respectively denote the number of nodes and features. To describe the specific connection relations among nodes, adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is defined that $\boldsymbol{A}_{ij}$ equals to 1 if node i and node j are directly connected, otherwise the value equals to 0. $\boldsymbol{D} \in \mathbb{R}^{m\times m}$ is the degree matrix of adjacency matrix $\boldsymbol{A}$. It is a diagonal matrix and its entry $\boldsymbol{D}_{ii}$ equals to $\sum_{j=1}^{m} \boldsymbol{A}_{ij}$. The computing formula of graph convolutional layer is defined in (1).

$$\boldsymbol{H}^{(l)} = \sigma(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}\boldsymbol{H}^{(l-1)}\boldsymbol{W}^{(l)}) \tag{1}$$

$\boldsymbol{H}^{(l)}$ is the hidden features of l^{th} layer and $\boldsymbol{H}^{(0)} = \boldsymbol{X}$. $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_m$ represents the adjacency matrix with added self-connections where $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ denotes identity matrix. The multiplication of $\tilde{\boldsymbol{D}}^{-1/2}$ is to perform normalization for $\tilde{\boldsymbol{A}}$. $\boldsymbol{W}^{(l)}$ denotes the learnable weight. $\sigma(\cdot)$ is the activation function. This formula is similar to the computing formula in fully connected neural networks that the added term $\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$ can help integrate the hidden features of different nodes based on connection relation in graph. Through multiple stacked graph convolutional layers, feature representation can be extracted based on relations among nodes for classification or regression task then.

3. THE PROPOSED MULTI-CHANNEL GRAPH CONVOLUTIONAL NETWORK

In this section, it is necessary to clarify the problem setting for end-point composition prediction of converter steelmaking first. Then, the proposed multi-channel GCN model will be described in details. There are two steps for the construction of the proposed model, namely sparse coding for graph structure learning and multi-channel network design.

3.1 The Problem Setting for End-point Composition Prediction of Converter Steelmaking

In this paper, with multiple process variables, the prediction of end-point element composition is concerned as the target of the proposed model. We use $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{m\times p}$ to respectively denote the data matrix of process variables and end-point element compositions where m, n and p are respectively the number of samples, process variables and element compositions. The proposed model aims to learn a mapping $f: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ so that the end-point element compositions can be predicted without direct measurement.

3.2 Sparse Coding for Graph Structure Learning

The first step of the proposed model is to construct a graph representing the correlations among element compositions. Because correlations among element compositions are considered in the proposed model, element compositions, rather than samples, should be taken as nodes to construct graph. With the relations among nodes previously unknown, Euclidean distance or kernel function has been used to calculate adjacency matrix to represent graph structure (Yan et al., 2016). In this way, general methods construct graph based on the distances among nodes rather than correlations, which may fail to explore some correlated variables far away from each other under distance index. Thus, we adopt sparse coding to explore the correlations among nodes.

Sparse coding aims at learning a dictionary and a sparse representation to sparsely represent data. Equation (2) is the mathematical form of the sparse coding objective function.

$$\min_{\boldsymbol{B},\boldsymbol{\alpha}_i} \sum_{i=1}^{m} (\|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1) \tag{2}$$

$\boldsymbol{x}_i$ and $\boldsymbol{\alpha}_i$ are the original feature vector and sparse representation of sample i, and m is the number of samples. $\boldsymbol{B} \in \mathbb{R}^{n\times k}$ is the dictionary matrix where n and k respectively denote the number of features and vocabulary. λ is the trade-off coefficient. $\|\cdot\|_1$ and $\|\cdot\|_2$ respectively denote L_1-norm and L_2-norm. The first term in objective function makes the sparse representation be able to reconstruct the original feature vector as similarly as possible. The second term in objective function is used to compress small entries to zero by L_1-norm.

Motivated by the idea of sparse coding, the matrix form of modified objective function and constraints are shown in (3).

$$\begin{aligned} &\min_{\boldsymbol{A}} \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{A}\|_2^2 + \lambda\|\boldsymbol{A}\|_1 \\ &s.t. \quad \boldsymbol{A} = \boldsymbol{A}^T \\ &\qquad\; A_{ii} = 0, A_{ij} \geq 0 \;\; (i,j = 1,2,...,m) \end{aligned} \tag{3}$$

Compared with the objective function in (2), the original feature vector and dictionary matrix are replaced by label matrix of element compositions in training set. The purpose of the replacement is to reconstruct the label matrix by itself. In addition to the constraint that all diagonal entries in $\boldsymbol{A}$ are zero, the entries of each column in $\boldsymbol{A}$ represent the constructional relation of each element by other elements. Thus, the solution for the objective function can well represent the correlations among element compositions and can be taken as the adjacency matrix of graph. Besides, to avoid the offset of positive and negative correlations which may influence normalization effect, it is supposed that the correlations should be equivalent and nonnegative. Sparsity can be implemented through L_1-norm term. The constrained minimization problem can be solved by convex optimization method.

Thus, with element compositions taken as nodes, it is feasible to apply sparse coding to learn the correlations among nodes and present the correlations into graph structure through adjacency matrix. With the learned adjacency matrix $\boldsymbol{A}$, to integrate this correlation information into the modelling through graph convolution, it is essential to design an appropriate network structure in the next step.

3.3 Multi-channel Network Design

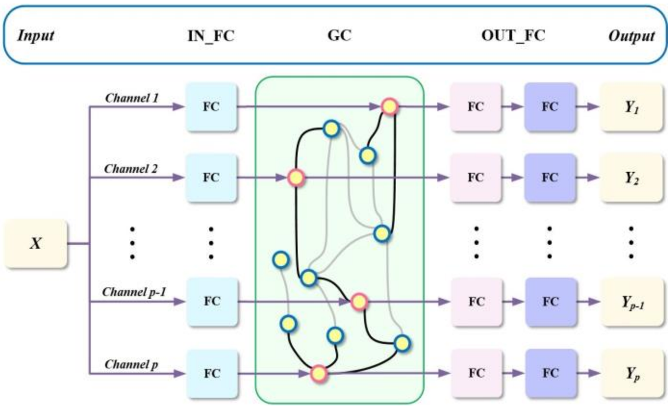

Fig. 1. The multi-channel network structure. Data flow are represented by horizontal purple arrows. The input data composed of process variables enter p independent channels where p is the number of element compositions. "IN_FC" and "OUT_FC" respectively denote input and output fully connected layer, and "GC" denotes graph convolutional layer. The IN_FC layer helps extract feature representation separately in different channels. Then, through a GC layer based on adjacency matrix learned by sparse coding, correlated channels will share feature representation (denoted by black solid lines) to integrate correlations among element compositions. The following OUT_FC layer can also learn feature representation separately in different channels and make predictions for different element compositions.

Most GCN models generally take samples as nodes in graph, so the computing formula in (1) can easily integrate features among samples. Now that element compositions are taken as nodes in our model, the correlations among them cannot be integrated by (1) directly. Thus, a multi-channel network is designed to help integrate the correlations among element compositions.

The multi-channel network structure is shown in Fig. 1. The input data are composed of multiple process variables, and p kinds of element compositions are the outputs of the network. The network consists of four stacked layers, including three fully connected layers and one graph convolutional layer. Firstly, input data separately enter p independent channels which represent different element compositions. The first fully connected layer can help learn feature representation in different channels separately. Then, with the adjacency matrix learned by sparse coding, the graph convolutional layer treats each channel as a node in graph and can fuse the features of correlated channels to integrate the correlations among element compositions. Through two following fully connected layers, features for different element compositions can be extracted in separate channels once again. After that, each channel will output the predictions for each kind of element composition. Through the alternate structure of fully connected layer and graph convolutional layer, the features will not only maintain independency in different channels, but also possess integrativeness from correlated channels. Thus, the design of this multi-channel network implements the integration of correlation information implied by adjacency matrix.

The fully connected layer is defined in (4).

$$\boldsymbol{H}^{(l)} = \sigma(\boldsymbol{H}^{(l\text{-}1)}\boldsymbol{W}^{(l)} + \boldsymbol{b}^{(l)}) \qquad (4)$$

$\boldsymbol{H}^{(l)}$ is the hidden features of lth layer and $\boldsymbol{H}^{(0)} = \mathbf{X}$. $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ respectively denote learnable weight and bias. In our model, ReLU function is selected as activation function $\sigma(\cdot)$. Besides, the graph convolutional layer is defined in (1) while the only difference is that we need to compute graph convolution across channels instead of samples. From input to output, the number of hidden neurons in four layers are respectively set as 1024, 256, 256 and 256 while in practice, the number of layers and hidden neurons can vary.

The loss function of multi-channel GCN is defined in (5). Then the learnable parameters in fully connected layers and graph convolutional layers can be trained by mini-batch gradient descent. $\boldsymbol{Y}, \hat{\boldsymbol{Y}} \in \mathbb{R}^{m\times p}$ respectively denote the ground truth and prediction of label matrix. To avoid overfitting, regularization term for graph convolutional layer is introduced into loss function with coefficient γ.

$$loss(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \sum_{i=1}^{m}\sum_{j=1}^{p}(\boldsymbol{Y}_{ij} - \hat{\boldsymbol{Y}}_{ij})^2 + \gamma\left\|\boldsymbol{W}^{(2)}\right\|_2^2 \qquad (5)$$

Through the sparse coding for graph structure learning and the design of multi-channel network, the correlations among element compositions can be integrated into multi-channel GCN prediction model for overall performance improvement. Sparse coding helps to explore the correlations among element compositions and present the correlation information into graph through adjacency matrix. Then, the design of multi-channel network can effectively utilize the correlation information in graph and share the feature representation of correlated elements through graph convolution across channels. Therefore, the proposed soft sensor model is able to capture the correlation information among element compositions.

4. ILLUSTRATION RESULTS

In this section, the proposed multi-channel GCN model is applied for the end-point element composition prediction of converter steelmaking. Experiment is performed based on dataset from real steelmaking process to validate its effectiveness. Table 1 is the overview of the dataset used in experiment.

Table 1. An overview of the dataset

Number of samples	2613
Number of process variables	35
Number of element compositions	12
Composition of process variables	The element compositions before alloy addition (12 variables)
	The amount of different kinds of alloy addition (14 variables)
	The temperature and weight of liquid steel (2 variables)
	Other data information about liquid steel (7 variables)
Element compositions	Als, C, Cr, Cu, Mn, Mo, Nb, Ni, P, S, Si, V

Because all of the variables are nonnegative and great scale difference exists among them, min-max normalization is performed to scale each variable into [0,1]. 80% of the samples are taken as training set and the rest are taken as test set. Experiments are repeated three times on different random splitting of dataset. The coefficient of determination (R^2) and root mean square error (RMSE) are taken as the evaluation criteria. The computing formulas of R^2 and RMSE are shown in (6) and (7).

$$R^2 = \frac{\sum_{i=1}^{m}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2} \tag{6}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{7}$$

m is the number of test samples. y_i and $\hat{y}_i$ respectively denote the ground truth and prediction of sample i. $\bar{y}$ is the average of the ground truth in test set. R^2 is a performance index smaller than 1 and represents the fraction of the total variation that is explained by the regression. RMSE represents the averaged deviation level between prediction and ground truth. The closer R^2 is to 1 and the smaller RMSE is, the better the model performs.

The trade-off coefficient λ in (3) and γ in (5) are set as 0.05 and 10, and the adjacency matrix is shown in Fig. 2. The batch size for training is set as 64 and Adam optimizer is used for parameter optimization while the learning rate is initially set as 0.001 and decays by half every 50 epochs. In experiment, the network trains for 200 epochs, and the loss curve in Fig. 3 shows the fast convergence during training.

In addition, PLS, SVR, fully connected neural network (FC), ELM and GCN are taken as contrast experiments. In SVR, RBF kernel function is used and tolerance margin is set as 0.05 while the penalty parameter C is optimized among [10^{-3},10^{-2},…,10^3] by grid search. As for FC, we construct a network composed of 2 hidden layers both with 256 neurons. ELM is single-hidden-layer structure and the number of hidden neurons are optimized among [2^5,2^6,…,2^{10}] by grid search as well. Because standard GCN also uses test samples during training, for a fair comparison, we respectively use training samples and test samples to construct graph for training and testing. RBF kernel function are used to learn sample graph and we construct a GCN consisting of 2 hidden layers both with 256 neurons.

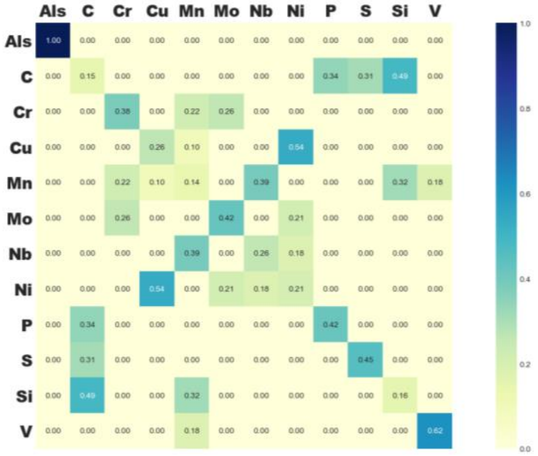

Fig. 2. The adjacency matrix. This is a symmetrical matrix and larger value represents stronger correlation. Values lower than 0.1 are set to zero in order to neglect some weak correlations.

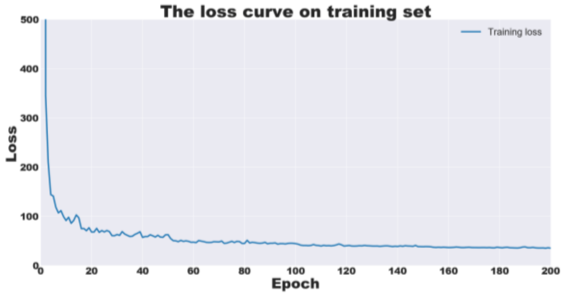

Fig. 3. The loss curve on training set.

The averaged results over three experiments are shown in Table 2 and the best performance is shown in bold. The averaged results over element compositions and averaged rankings (No.1 means the best and No.6 means the worst) among six models are shown in last two rows of the table.

From experiment results, it is obvious to find that multi-channel GCN outperforms other five models on half of the element compositions and performs best in average. Although other models perform better on some element compositions, the performance differences are pretty small so that multi-channel GCN has pretty prominent performance in ranking. Besides, the performance of different models shows some similarities that most models perform quite well on element C, Cr, Cu, Mn, Mo, Nb, Ni, S and Si while performing badly on element Als, P and V. It can be seen that multi-channel GCN performs much better than other models on these hard-to-predict elements, especially on element V. Because of the integration of correlation information in multi-channel GCN, the learning for different element compositions can cooperate with correlated element compositions so that the performance on hard-to-predict elements will improve. But not all element compositions are correlated with others. From the adjacency matrix in Fig. 2, element Als is an isolated node without connection to other element compositions in graph. Thus, the learning for element Als is independent from others which also reveals in the poor performance on element Als. However, some hard-to-predict elements correlated with others will perform better with the help of correlation information which shows the advantage of proposed model. As for other contrast models, PLS performs close to SVR, and FC performs close to ELM. They perform comparably on easy-to-predict elements while differences mainly lie on hard-to-predict ones. But GCN performs poorly even on some easy-to-predict elements because the sample graph is different between training and testing.

Table 2. Experiment results of different models

	Evaluation Criterion: R^2						**Evaluation Criterion: RMSE**					
Elements	**PLS**	**SVR**	**FC**	**ELM**	**GCN**	**Multi-channel GCN**	**PLS**	**SVR**	**FC**	**ELM**	**GCN**	**Multi-channel GCN**
Als	0.121	0.314	0.156	-0.239	-0.951	**0.331**	0.0438	0.0387	0.0420	0.0495	0.0605	**0.0382**
C	0.830	**0.890**	0.878	0.826	0.759	0.880	0.0202	**0.0165**	0.0174	0.0203	0.0244	0.0173
Cr	0.929	0.900	0.914	**0.949**	0.625	0.938	0.0194	0.0236	0.0214	**0.0167**	0.0454	0.0185
Cu	**0.966**	0.925	0.947	0.932	0.792	0.952	**0.0046**	0.0069	0.0058	0.0059	0.0105	0.0055
Mn	0.903	0.943	0.783	**0.949**	0.515	0.935	0.1303	0.0996	0.1683	**0.0947**	0.2902	0.1067
Mo	0.964	0.816	0.927	0.931	0.873	**0.970**	0.0060	0.0139	0.0085	0.0075	0.0114	**0.0056**
Nb	0.904	0.917	**0.920**	0.901	0.641	0.911	0.0041	0.0038	**0.0037**	0.0041	0.0079	0.0039
Ni	0.966	0.946	0.973	0.935	0.844	**0.979**	0.0040	0.0051	0.0036	0.0048	0.0085	**0.0031**
P	0.728	0.622	0.280	0.620	-0.132	**0.729**	0.0019	0.0022	0.0028	0.0022	0.0038	**0.0019**
S	0.796	0.806	0.771	0.855	0.083	**0.871**	0.0030	0.0031	0.0032	0.0026	0.0067	**0.0025**
Si	0.873	**0.897**	0.795	0.893	0.603	0.875	0.0331	**0.0298**	0.0392	0.0302	0.0581	0.0325
V	0.223	0.374	0.400	0.104	0.262	**0.531**	0.0115	0.0103	0.0101	0.0121	0.0112	**0.0088**
Mean	0.767	0.779	0.729	0.721	0.410	**0.825**	0.0235	0.0211	0.0272	0.0209	0.0449	**0.0204**
Ranking	3.33	3.08	3.50	3.58	5.75	**1.75**	3.25	3.33	3.50	3.42	5.75	**1.75**

To prove the effectiveness of sparse coding for correlation description, we compare it with Pearson correlation coefficient and RBF kernel function, and the averaged results are shown in Table 3. RBF kernel function learns the graph structure based on the distances among nodes, so the correlations among element compositions are not involved and the poor performance proves it. Pearson correlation coefficient is an index describing linear correlation strength, while these coefficients cannot reconstruct node features through linear combination of others, which is the property of convolution calculation. Thus, sparse coding learns the constructional correlations among nodes, and combines with GCN best.

Table 3. Different methods for graph structure learning

Methods for graph structure learning	R^2	RMSE
Sparse coding	**0.825**	**0.0204**
Pearson correlation coefficient	0.799	0.0209
RBF kernel function	0.778	0.0224

These experiment results show that the performance of multi-channel GCN featuring the integration of correlations among element compositions is pretty effective. Compared with other general prediction models, multi-channel GCN adopts graph structure to describe the correlations among element compositions and uses graph convolution across channels to integrate these correlations. While other general models can get comparable performance on easy-to-predict elements, the superiority of multi-channel GCN mainly reveals on those hard-to-predict elements and the outstanding comprehensive performance. Besides, we show the excellent performance of sparse coding and its suitability with GCN.

5. CONCLUSIONS

In this study, a multi-channel GCN model is proposed for the end-point element composition prediction of converter steelmaking which can effectively integrate correlations among element compositions. Through two modelling steps, the correlations among element compositions can be first described in graph structure by sparse coding and then integrate into the model by multi-channel network design. Experiment results based on real dataset in converter steelmaking demonstrate the effectiveness and superiority of the proposed model. It is the first time that the GCN model is successfully reported for soft sensing in industrial field. Future work will focus on how to effectively improve the performance of prediction model on hard-to-predict elements when there are weak variable correlations.

ACKNOWLEDGEMENT

This work is supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (No.U1709211), Zhejiang Key Research and Development Project (2019C03100 and 2019C01048), and Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies Foundation.

REFERENCES

Atwood, J. and Towsley, D. (2016). Diffusion-Convolutional Neural Networks. *Advances in neural information processing systems*, 1993-2001.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral Networks and Locally Connected Networks on Graphs. *arXiv preprint*, arXiv:1312.6203.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *Advances in neural information processing systems*, 3844-3852.

Dong, X. and Dong, S. (2014). The Converter Steelmaking End Point Prediction Model Based on RBF Neural Network. *Applied Mechanics and Materials*, 577, 98-101.

Han, M. and Liu, C. (2014). Endpoint prediction model for basic oxygen furnace steel-making based on membrane algorithm evolving extreme learning machine. *Applied Soft Computing*, 19, 430–437.

He, F. and Zhang, L. (2018). Prediction model of end-point phosphorus content in BOF steelmaking process based on PCA and BP neural network. *Journal of Process Control*, 66, 51-58.

Kipf, T. N. and Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint*, arXiv:1609.02907.

Lecun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. E., and Jackel, L. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541-551.

Li, C., Zhao, H., Xie, X., and Yin, Q. (2011). Prediction of end-point phosphorus content for BOF based on LM BP neural network. *Iron and Steel/ Gangtie*, 46(4), 23-25.

Liu, D., Zou, Z., and Yu, A. (2007). Prediction Model of End-point Phosphorus Content in BOF Process. *China Metallurgy*, 17(1), 34-37.

Micheli, A. (2009). Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Transactions on Neural Networks*, 20(3), 498-511.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2020). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 1-21.

Xu, L., Li, W., Zhang, M., Xu, S., and Li, J. (2011). A model of basic oxygen furnace (BOF) end-point prediction based on spectrum information of the furnace flame with support vector machine (SVM). *Optik*, 122(7), 594-598.

Yan, W., Guo, P., Tian, Y., and Gao, J. (2016). A Framework and Modeling Method of Data-Driven Soft Sensors Based on Semisupervised Gaussian Regression. *Industrial & Engineering Chemistry Research*, 55(27), 7394-7401.

Yu, Z. (2001). Contemporary converter steelmaking technology. *Steelmaking*, 17(1), 13-18.