Chemical Engineering Science 225 (2020) 115796

Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Operational trend prediction and classification for chemical processes: A novel convolutional neural network method based on symbolic hierarchical clustering

Yongjian Wang [a,b], Yichi Zhang [b], Zhe Wu [b], Hongguang Li [a,*], Panagiotis D. Christofides [c,**]

[a] College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
[b] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA
[c] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

HIGHLIGHTS

- Operational Trend Prediction and Classification for Chemical Processes.
- Convolutional Neural Network Method Based on Symbolic Hierarchical Clustering.
- Application to a Chemical Plant via Industrial Data.
- Favorable Comparison with Traditional Neural Network Methods.

ARTICLE INFO

Article history:
Received 31 January 2020
Received in revised form 4 May 2020
Accepted 10 May 2020
Available online 22 May 2020

Keywords:
Operation trend
Convolutional neural network
Symbolic hierarchical clustering

ABSTRACT

In modern industrial chemical engineering plants, the quality of the product is closely related not only to the process design but also to the efficiency of human operation. Currently, single-step prediction models are adopted by process engineers to estimate the immediate system response. However, those single-step prediction models are limited as they don't enable the operator to visualize the complete series of effects associated with the operation in the long run. In order to help make prescient predictions, this paper proposes a novel symbolic hierarchical clustering (SHC) based convolutional neural network (CNN) method for trend prediction and classification. Firstly, the raw historical operation data series are symbolized from numerical values to strings according to their distinct characteristics. Secondly, the hierarchical clustering method is used to eliminate the low-frequency operation trends and to determine and label the types of operational trends for the symbolized dataset. Subsequently, the categorized dataset and its respective label are fed into a specially tailored CNN for the training of the CNN model for trend classification. Finally, to demonstrate the effectiveness of the proposed SHC-CNN algorithm, the proposed method is applied to the methanol production process of Hainan Petrochemical Co., Ltd. to predict and classify its main operational trends. In addition, the superiority of SHC-CNN operational trend prediction is demonstrated through the comparison with traditional neural networks.

1. Introduction

In the modern industrial chemical processes, increasingly stringent product quality demands lead to more and more sophisticated manufacturing process designs (Boguslavskii and Kirsanov, 1989), for which manual operations from process engineers are often needed to handle possible process disturbances and guide the normal plant operation. Due to the complexity of chemical plants, the aforementioned operations require an abundant prior knowledge and extensive experience of the processes, which can affect the efficiency and effectiveness of the overall manufacturing mechanism and the quality of the final product (Wang and Li, 2018). To ensure that the human interventions will lead to positive influences on the entire production process and less empirical knowledge is required for process engineers, it is beneficial to develop a process model and use it to guide the decision making.

Process models are generally developed using first-principles knowledge (mechanism-based model) or data-driven approach (data-driven model) (Tong et al., 2015). Specifically, the

* Corresponding author.
** Corresponding author.
E-mail addresses: lihg@mail.buct.edu.cn (H. Li), pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.ces.2020.115796
0009-2509/

first-principles models are developed based on the physical knowledge of the process. For example, the species, momentum, and energy transport phenomena, and the reaction kinetics are involved in the modeling of a chemical engineering plant. Researchers have delved in utilizing a first-principles model in a variety of fields, establishing accurate first-principles models to represent the physical characteristics of processes.

However, the construction of first-principles models face various challenges. First, important process parameters in the actual production process may remain difficult to obtain. Also, it is often impossible to derive accurate mathematical equations between process parameters and outputs. As a result, researchers have begun to investigate and adopt an alternative modeling method, data-driven modeling. Data-driven modeling is a data-based modeling approach, which utilizes a variety of numerical algorithms to abstract and extract the key information and the input–output relationships from the historical database (Smarra et al., 2018; Kim et al., 2018). In the modern industrial processes, the maturity of sensor and storage technologies such as distributed control systems (DCS) enable the generation of a large historical database. This large database serves as the basis for the data-driven model formulation, because the comprehensive features of the production processes are preserved in the rich historical data set, from which, representative models can be obtained to accurately describe the industrial processes.

To construct a statistical model based on the data set, machine learning method is adopted, which is a branch of the field of artificial intelligence that uses a variety of mathematical algorithms to analyze the data set for regression and classification in order to make future predictions (Dunjko and Briegel, 2018). For example, Amasyali and El-Gohary (2018) offered a review of the currently developed data-driven models that predicted building energy consumption in industry, and highlighted the future research directions in the area. Li et al. (2018) adopted a support vector machine (SVM) method to construct a data-driven model to characterize the non-explicit relationship from the operating data, which resulted in a good classification and a reasonable computation time in modeling the air-cooling condenser process. Yu et al. (2018) proposed a data-driven model based on Fourier transform and support vector regression for a precise monthly reservoir inflow forecasting. Ding et al. (2019) constructed a multi-scale data-driven model to characterize the thermal atomic layer deposition process and to make predictions for the optimal operating conditions.

Among the large realm of machine learning methods, deep learning has recently drawn a lot of attention due to the increasing computational power and its unparalleled advantage in extracting hidden information in complex data sets (Zhao et al., 2019). Deep learning network consists of the following popular forms of neural networks: convolutional neural network (CNN) (Krizhevsky et al., 2012), recurrent neural network (RNN) (Lukoševičius and Jaeger, 2009), and deep belief network (DBN) (Mohamed et al., 2011). Within those structures, CNN is widely recognized and applied for its ability to extract deep and correlational features. CNN is a feedforward neural network with convolutional layers, often involving deep network structures (Wang et al., 2019). A lot of scholars have focused on the theory and application of CNN in recent years. For example, LeCun et al. (2015) first introduced CNN to classify the handwriting, which became the most generic CNN structure adopted in future studies. Lu et al. (2019) used parallel network structure to increase the classification accuracy of pathological brain detection, which is known as the AlexNet, whereas GoogleNet used a deeper network structure and a higher number but smaller scale convolution kernels to achieve higher precision (Szegedy et al., 2015). VGG, born out of the need to minimize the number of parameters in the CONV layers and improve on training time, inherited and improved the deep multi-convergence features of GoogleNet (Simonyan and Zisserman, 2014). CNN structures have the ability to learn and to classify the input information according to its hierarchical structure. Specifically, CNN accomplishes the extraction and fusion of specific features through the use of sliding window techniques and enables the classification of the labeled results through the dense classification links. In our work, the operation trend prediction requires the selected machine learning method to be able to recognize and extract the hidden operational features out of the complex historical data set. Thus, CNN is considered suitable for our purpose and is used to handle operational prediction problems.

To develop a supervised machine-learning model that learns and classifies the operational trends, we first need to extract the types of operational trends from historical data. Specifically, we adopt the clustering method for the training data series clustering as we want to classify the similar operational trend as the same category. Clustering is a process that divides a collection of physical or abstract objects into multiple subsets consisting of similar objects, which preserves the main features of the data, thereby greatly reducing the difficulty of feature extraction (Jain et al., 1999). Commonly used clustering methods for time-series data are: segmentation clustering (Iwahashi et al., 2018), hierarchical clustering (Murtagh, 1983), density clustering (Bryant and Cios, 2018), model clustering (Ding et al., 2018) and grid clustering (Xu et al., 2018).

Nevertheless, because it is impossible to determine the types of clusters in advance, the classification of operational trends in industrial processes cannot adopt simple segmentation clustering methods. Additionally, model clustering and grid clustering methods are not suitable due to the complex characteristics of the process industry. Moreover, since the dense steady-state in the data set is very close, it is difficult to be separated by the density clustering method. Therefore, in this paper, we use the hierarchical clustering method to classify the operation trend data series, in which the historical operation data set is categorized according to the learned trend types. The basic idea behind the hierarchical clustering method is to calculate the similarity between nodes with a similarity checking criterion, and then the nodes can be sorted from high to low according to the calculated similarity. Finally, the nodes are reconnected and regrouped according to the classification criterion. Hierarchical based clustering methods have been extensively studied by various researchers. Dasgupta framed a similarity-based hierarchical clustering as a combinatorial optimization problem, which showed the beneficial attributes of the resulted cost function (Cohen-Addad et al., 2019). Lu et al. (2018) proposed a novel data clustering algorithm, which did not require the number of clusters as the input parameter, allowing the user to conveniently acquire the proper number of clusters. In addition, Peterson et al. (2018) presented a hybrid non–parametric clustering approach, which could find the general shapes and structures in data sets.

Despite the popularity of the neural network trend prediction models, the established models only allowed prediction of one sampling step, which did not enable the operator to have a far-sighted and prescient view of the future outcome of the operation. Motivated by the above considerations, this paper proposes a convolutional neural network operation trend prediction and classification method, based on symbolic hierarchical clustering (SHC-CNN). First, the historical data of the industrial process are collected, which are then symbolized and classified with the hierarchical clustering method. The obtained operational trend data and group labels are fed into CNN for classification prediction training. In order to demonstrate the validity and effectiveness of the proposed method, we apply it to the methanol production process of Hainan Petrochemical Co., Ltd., and the result is compared

with that of the traditional CNN and RNN methods. The rest of this paper is structured as follows: Section 2 explains the symbolization method, the hierarchical clustering method, and the convolutional neural network. Section 3 describes the construction of the integrated SHC-CNN operation trend prediction and the classification method. In Section 4, the developed SHC-CNN algorithm is applied to the methanol production process of Hainan Petrochemical Co., Ltd., and the results are compared with other traditional classification methods.

2. Preliminaries

The SHC-CNN operation trend prediction for a complex industrial process consists of the following three aspects: symbolization, hierarchical clustering and convolutional neural network. In this section, we will introduce the underlying theory and the basic structure of each of the aforementioned methods.

2.1. Symbolization

Symbolization techniques convert raw numerical data to spatial data with classification, simplification and exaggeration. From the symbolized results, various patterns can be extracted according to the distinct characteristics, relative importance, and related positions. Symbolic aggregate approximation (SAX) algorithm is a common method for time-series data symbolization, which creates a discretized symbolization pattern according to the magnitude of the time-series data points, and then transforms each raw data into its respective symbolic form. Thus, it helps solve problems which involve data sets that cannot be easily visualized, by extracting the implicit patterns. SAX usually involves a two-step transformation.

(1) In time-series standardization, we normalize the time-series with respect to the mean and standard deviation:

$$x^* = \frac{x-\mu}{\sigma} \tag{1}$$

where x represents the original data, x^* represents the standardized data, μ and σ represent the mean and standard deviation of the original data, respectively. Specifically, the original time-series data we considered in the simulation example of this manuscript satisfies Gaussian distribution on the vertical axis. As a result, it will be normalized through Eq. (1) to a standardized Gaussian distribution.

(2) In time-series data segmentation, we split a piece of time data of length n into w equal length blocks, and obtain a series of blocked data $\overline{S} = [\overline{s_1}, \ldots, \overline{s}_w]$, with the i_{th} data point as the average of data points in i_{th} block:

$$\overline{s_i} = \frac{w}{n} \sum_{j=\frac{n(i-1)}{w}+1}^{\frac{n\cdot i}{w}} x_j, \qquad 1 \leqslant i \leqslant w \tag{2}$$

where x_j represents a single sample point of time-series data, j represents the serial number of the block, n is the integer multiple of w, and w stands for the length of the time series data. After the standardization process, the time-series data will be redistributed on the axes.

Fig. 1 shows an example of data symbolization under SAX. It is demonstrated that the vertical axis is divided into three regions according to our classification criterion, and thus, the time-series data can be equally distributed into the respective region. Subsequently, the fluctuating time-series data is converted to a string of characters *baabccbc*// under SAX. Therefore, the SAX algorithm is able to convert any length of time-series data into a corresponding string, allowing for easier computation and achieving good noise elimination.

2.2. Hierarchical clustering

Hierarchical clustering is a type of clustering algorithm, which creates a hierarchical nested clustering tree by calculating the similarity between different types of data points. In the clustering tree, the lowest layer consists of the original data points of different categories, and the top layer of the tree consists of the cluster root node. There are two methods for clustering tree creation: a bottom-up merging method and a top-down splitting method. The top-down split-level clustering method treats all samples as one cluster, which are then iteratively divided into smaller clusters until only one sample is left in each cluster. The bottom-up condensed hierarchical clustering treats each sample in a different cluster, and the nearest pair of clusters are merged until all the samples belong to the same cluster. clustering method can be constructed as follows: (1) Obtain the number of the samples, which is the number of initial clusters. (2) Calculate the samples merging according to the similar metrics between samples. (3) Combine the samples that satisfy the similarity criterion into one cluster. (4) Repeat Step (1), (2) and (3) until the historical data is condensed into one cluster. Three kinds of similarity metrics are usually involved in the hierarchical clustering; which are given as follows:

(1) Single chain: The distance between the clusters that is the closest.

$$D_{\min} = \min_{x \in C_m, z \in C_n} dist(x,z) \tag{3}$$

(2) Full chain: The distance between the clusters that is the farthest.

$$D_{\max} = \max_{x \in C_m, z \in C_n} dist(x,z) \tag{4}$$

(3) Average chain: The average distance between clusters.

$$D_{\text{average}} = \operatorname*{average}_{x \in C_m, z \in C_n} dist(x,z) = \frac{1}{|C_m|}\frac{1}{|C_n|}\sum_{x \in C_m}\sum_{x \in C_n} dist(x,z) \tag{5}$$

where C_m and C_n represent two different kinds of clusters, and x and z are the data points that are contained in each of the clusters, respectively.

2.3. Convolutional neural network (CNN)

As discussed in the introduction, CNN is a feedforward neural network with convolutional layers and is able to learn and classify the input information according to its hierarchical structure. Here, we introduce the most classical CNN structure. Fig. 2 shows the general form of the CNN structure that involves the convolutional layers and sub-sampling layers repetitively, which are the two major component layers of CNN. Additionally, parameter sharing and sparse connectivity are the two distinctive features of network connection. These important features will be explained in more details in the following subsection.

2.3.1. Convolutional layer

Convolutional layers are introduced in CNN for the extraction of correlational feature and reduction of the size of the computational matrix. Each convolutional layer in a CNN is obtained by a group of filters, and the dimension of a convolutional layer is equal to the number of filters. Depending on the characteristic of the filter,

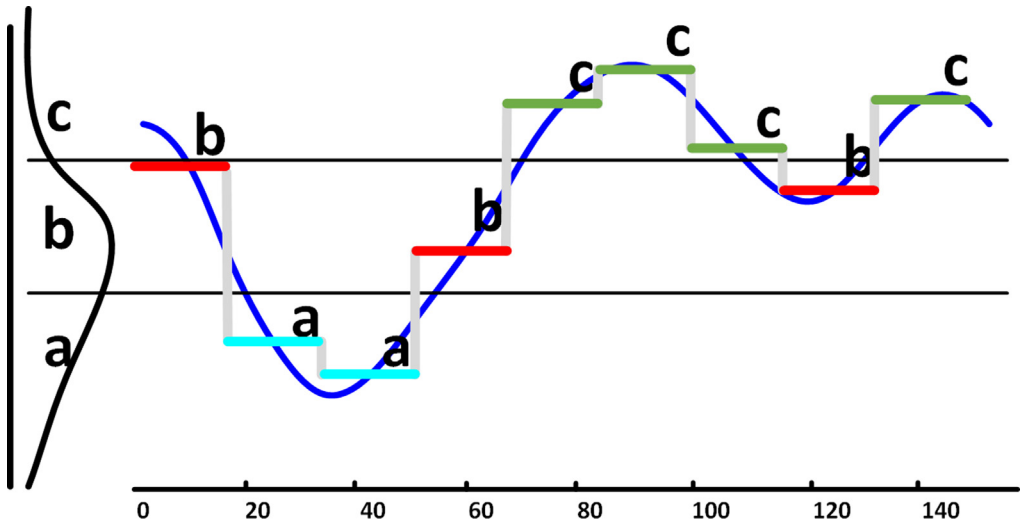

Fig. 1. Schematic of SAX method, where the blue curve represents the data samples, and a, b, c represent the symbolized variables. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

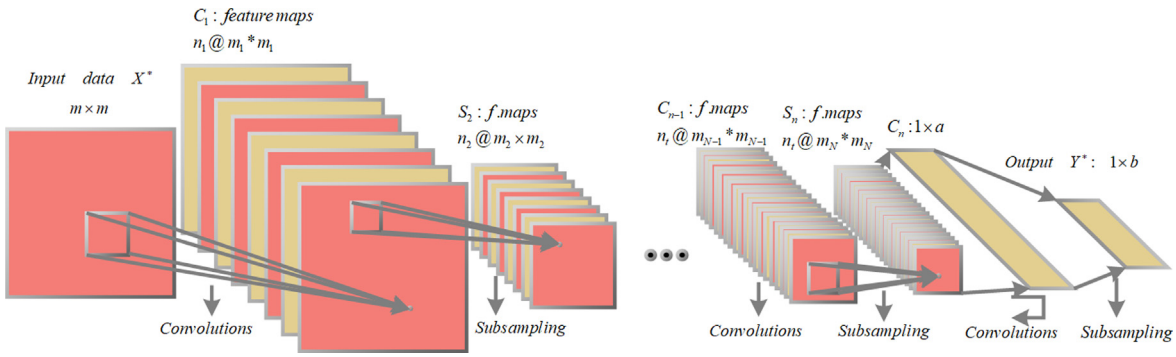

Fig. 2. Convolutional neural network structure showing that the first convolutional results C_1 with n_1 feature maps of dimension $m_1 * m_1$ are obtained after the first convolution layer, followed by subsampling, convolution and pooling processes, where X^* represents the input data of dimension $m * m$, S_1 represents the results of subsampling with n_2 feature maps of dimension $m_2 * m_2$, C_n represents the $1 * a$-dimensional data after flattening, and Y^* of dimension $1 * b$ represents the final output.

the corresponding convolution operation can extract different input features, and therefore, multiple distinctive convolution kernels can be used together to process the same input object and to extract its different features. The convolutional output can be obtained using the weight parameter and convolution kernels. It is demonstrated that the convolutional kernels have much smaller dimensions than the original input layer, which substantially reduces the computational cost compared to processing everything together with a dense layer. After the information is passed through the convolutional layer, the results are sent to the pooling process.

2.3.2. Pooling layer

The results from the previous convolutional layer are transferred as input to the following pooling layer, which downsamples the feature maps by summarizing the characteristics of features according to the given criterion in patches of the feature map. In order to process the upstream information, the neurons of the pooling layer are connected to the local accepting domain of the input layer, while different receiving regions do not overlap. Also, the characteristics of a pooling kernel correspond to the features of the connected convolutional kernel. Thus, according to the translation of the pooling layer, the pooling layer can distinctly perform quadratic feature extraction on its local accepting domain. Commonly used pooling methods include maximum pooling, average pooling, and random pooling. Among the three methods, maximum pooling and average pooling are considered in this paper.

2.3.3. Sparse connection

The sparse connection, also known as sparse interaction or sparse weight, is an important feature of CNN that considerably improves the computational efficiency. This concept of non-dense connection roots from neuroscience, where each visual cell in the human eye is only sensitive to a small portion of the entire retina area. In the dense neural network layer connection, a neuron in the input layer is connected to each pixel in the given input layer, and the same is true for every other neuron. In contrast, in the sparse connected neural networks, each of the neuron is only connected to a portion of the pixels. This localization reduces the number of weights need in the neural network, which not only cuts down the need for storage but also speeds up the matrix calculation in each layer. One of the most popular applications of CNN is in image processing domain, particular when dealing with an image of a large dimension, with millions of pixels. In that case, each convolutional and pooling kernel only take in less than ten pixels to detect small and meaningful features, such as the contour of an object. As a result, instead of processing and storing millions time millions of weights in each layer, the sparse connection reduces the computational cost dramatically.

2.3.4. Weight sharing

Weight sharing refers to the extraction process of a feature using the same filter. It is demonstrated that in a feed forward neural network, each element of the weight matrix is used for every calculation of the element in the output layer, where lines with different colors represent different kinds of parameters. Nevertheless, in the weight sharing networks, each element of the kernel only acts on its local inputs. This weight sharing feature in the convolution operation enables the network to only learn a set of parameters without the need to explore a separate set of parameters for each location. Although this feature does not change the run time of the forward propagation, the storage requirements of the model can be significantly reduced since a significantly smaller amount of the parameters is required compared to the dense neural networks. Therefore, the weight sharing mechanism facilitates the multiplication of dense matrices with simpler storage requirements and improved computational efficiency.

3. Integrated CNN operational trend prediction and classification method

As introduced in Section 1, the proposed industrial operation trends prediction method has the ability to learn from the historical trend of the process and thus to guide the operator to control the whole process system in a more effective and efficient way, which will lead to higher productivity for industrial plants. If we can extract valuable information from the operation data, we could establish this model based on the extracted operation modes. Fortunately, DCS can keep a huge amount of operation data in time-series, which could help the operators, even without rich experience, to maintain the whole process to always work under optimal operating conditions. In this section, we will explain in detail the integrated construction of a novel operation trend prediction and classification method based on symbolic hierarchical clustering and the CNN structure (SHC-CNN) that has been discussed in Section 2. The integrated process of the proposed convolutional neural network method based on symbolic hierarchical clustering (SHC-CNN) is illustrated in Fig. 3.

3.1. Symbolic hierarchical clustering

Utilizing the SAX method discussed in Section 2, the original time-series data can be processed into a one-dimensional form of a string, with each character preserving the numerical and process time information. The time-series data symbolization can reduce the storage requirement and the computational time. Additionally, it is demonstrated that this division of the time-series numerical interval is a better reflection of the intrinsic historical trend, which greatly enhances the effectiveness of the classification and prediction.

There are two important parameters in the implementation of the SAX algorithm: the number of divisions needed and the division criterion. In this paper, the number of divisions is nominally assumed to be q, which should be determined in the real industrial manufacturing process according to the need for the operation. In addition, a Gaussian distribution is demonstrated to provide an appropriate division criterion, as the normalized distribution ensures the unbiased division of the continuous data point. As discussed in the normalization step of Eq. (1), our post-processed time-series data follows the standardized Gaussian distribution. As a result, the vertical axis is divided into q intervals with the Gaussian distribution probability "breakpoint", where $p_1, p_2, \ldots, p_{q-1}$ are the "breakpoints" and the area of the interval in the Gaussian distribution $N(0,1)$ between two adjacent "breakpoints" is $1/q$. The "breakpoints" determined is shown in Table 1.

Utilizing the Gaussian distribution division criterion for q types of divisions, we can convert the raw numerical data set into sets of strings with the SAX algorithm. In order to visualize the industrial data set segmentation, we take the SAX results of a selected period of sequence data as an example. As shown in Fig. 4, it is a piece of industrial process data that is standardized and divided into five intervals. We can learn from Table 1 that the four "breakpoints" are: -0.84, -0.25, 0.25, and 0.84. The five sections have the same area in the Gaussian distribution, where the interval smaller than -0.84 is labeled as a, and the others are labeled as b, c, d, and e as follows. In this fashion, the entire sequence of the data can be dissected and classified into discrete symbolized time-series form from 1 to k. In addition, it should be noted that the proposed method is not restricted to data from Gaussian distribution. To address the time-series data with non-Gaussian characteristic, we can adopt other normalization methods (e.g., dispersion normalization for data of positive distribution such as the exponential distribution and Poisson distribution). Thus, the statistical information can then be used to determine the distribution probability break-point following the similar approach.

$$S = \{S_1, S_2, \cdots, S_k\} \tag{6}$$

After the raw numerical data series are transformed into string series via SAX, we need to determine the number of types in order to learn and classify the symbolized data set. As introduced in Section 2, we use the bottom-up condensed hierarchical clustering method to merge similar clusters together. To quantitatively determine the similarity between the string series, the Levenshtein distance method is used, which utilizes the minimum number of edit operations required between two strings, from one to another. Assuming that there are two sub-sequences S and T, we can calculate the Levenshtein distance $lev_{S,T}$ between the two time-series data as follows:

$$lev_{S_i,T_j} = \begin{cases} D_{max_{i,j}}, & if \quad D_{min_{i,j}} = 0 \\ \min \begin{cases} lev_{S_{i-1},T_j} + 1 \\ lev_{S_i,T_{j-1}} + 1, \\ lev_{S_{i-1},T_{j-1}} + 1_{(S_i \neq T_j)} \end{cases} & otherwise \end{cases} \tag{7}$$

where S_i stands for the i_{th} letter of the string S, T_j stands for the letter of the string T, and $1_{S_i \neq T_j}$ stands for a Dirichlet function, representing that the value equals 1 when $S_i \neq T_j$, and equals 0 when $S_i = T_j$. The i and j are the indices for letters in each string, and $D_{min_{i,j}}$ refers to the minimum distance between the i and j character. $D_{max_{i,j}}$ refers to the maximum distance between the i and j character. S and T stand for two different sub-sequences, respectively, which are obtained through the symbolization method mentioned in the previous section. Utilizing this equation, we will calculate the distance between all of the sub-sequences to build a clustering tree. The similarity η between two strings, S and T, can be calculated as follows after the Levenshtein distance is obtained:

$$\eta = 1 - \frac{lev_{S,T}(i,j)}{S_{length}} \tag{8}$$

where S_{length} stands for the length of the string S. In this work, in order to make the calculation result to be comparable easily with the correct dimension, we divide the strings into subsets with equal lengths. In the hierarchical clustering process, every single data can be regarded as a cluster at first. The Levenshtein distance matrix consists of one row per node and one column per node and contains all the pairwise distances. Metric calculations between samples are performed according to the distance. The samples with the smallest distance are condensed into one cluster, and then the distance matrices in the clusters are updated after condensing. This process is repeated until it can no longer continue to condense. The cluster-

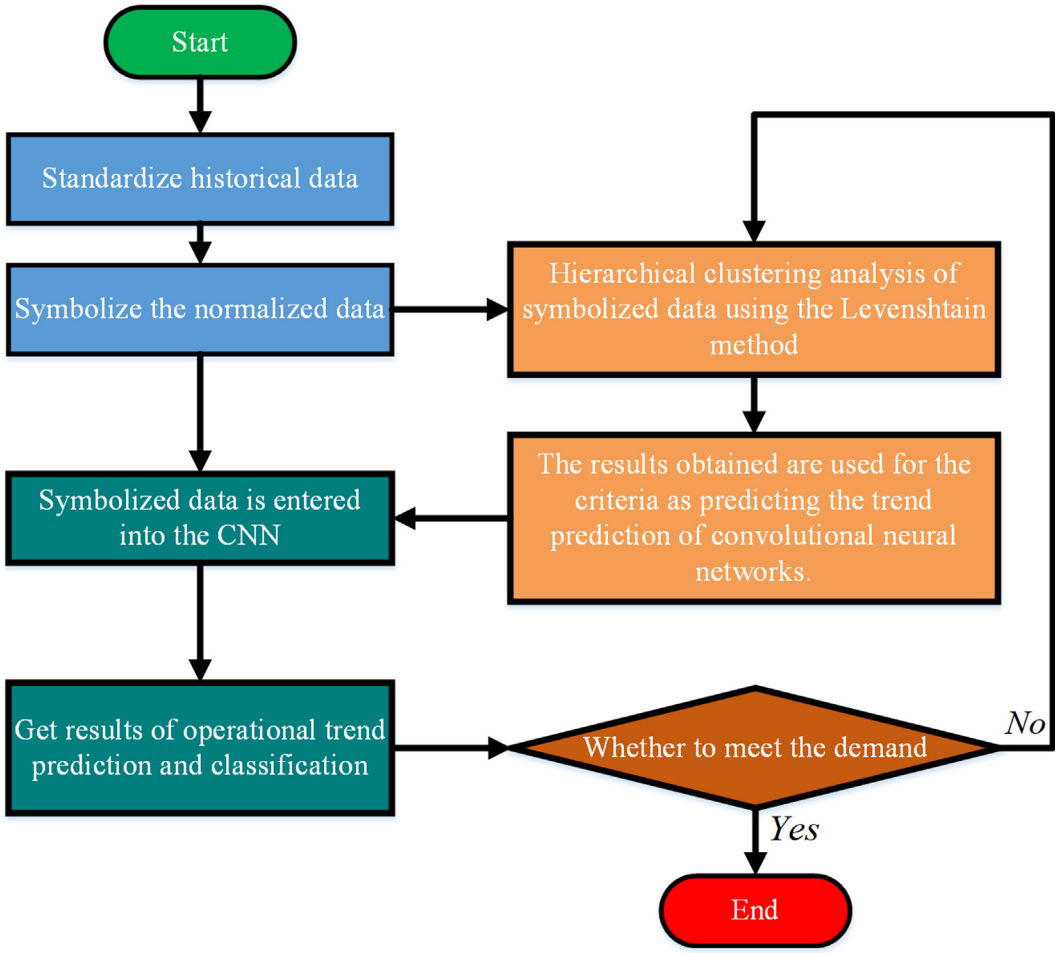

Fig. 3. Workflow of the proposed SHC-CNN method.

Table 1
Breakpoint division of the SAX method for data symbolization.

q	3	4	5	6	7	8	9	10
p_1	−0.43	−0.67	−0.84	−0.97	−1.07	−1.05	−1.22	−1.28
p_2	0.43	0	−0.25	−0.43	−0.57	−0.67	−0.76	−0.84
p_3		0.67	0.25	0	−1.08	−0.32	−0.43	−0.52
p_4			0.84	0.43	0.18	0	−0.14	−0.25
p_5				0.97	0.57	0.32	0.14	0
p_6					1.07	0.67	0.43	0.25
p_7						1.15	0.76	0.52
p_8							1.22	0.84
p_9								1.28

ing results are shown as a tree diagram in Fig. 5. In Fig. 5, we use the data presented on the Fig. 4 to make the description more consistent. The letters represent the symbolized time-series data, and the dashed lines of different colors represent different maximum similar distances, respectively, where the time-series below each dashed line can be considered similar. Different distances will lead to different clustering results. For example, if we set the maximum similar distance as the purple dotted line, we will get two different kinds of categories (see Fig. 6).

3.2. The SHC-CNN algorithm

In order to perform the operation trend prediction, an input–output model is developed using the aforementioned CNN structure. Specifically, the category of classification can be obtained with the hierarchical clustering method mentioned in Section 3.1, and the data series and the respective labels are fed into the CNN for classification learning and prediction. The constructed CNN has the data flow as shown in Fig. 3. The symbolized input variables X^* first pass through a convolution link, and then, the results of the convolution are sent to the pooling link. The number of convolution kernels can be obtained by grid search method based on the training and validation accuracy. The convolution and pooling operations are repeated m^* times, where the value of m^* can be obtained from the actual industrial process data by a random trial and error method. The magnitude of m^* is selected as small as possible to reduce the computation costs and avoid the possibly associated over fitting. After the consecutive convolutional and pooling operation, the results of the last pooling P^* are flattened in the form of a full convolution, and then the network's trend prediction

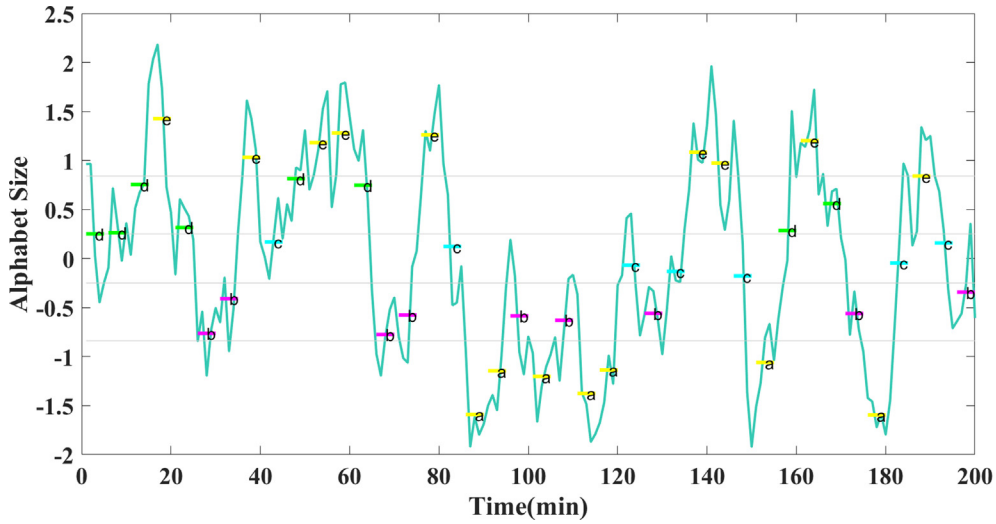

Fig. 4. Illustration of time-series data symbolization, where the blue profile represents the historical data and the red profile with letters "a, b, c, d, e" represent the symbolization result. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

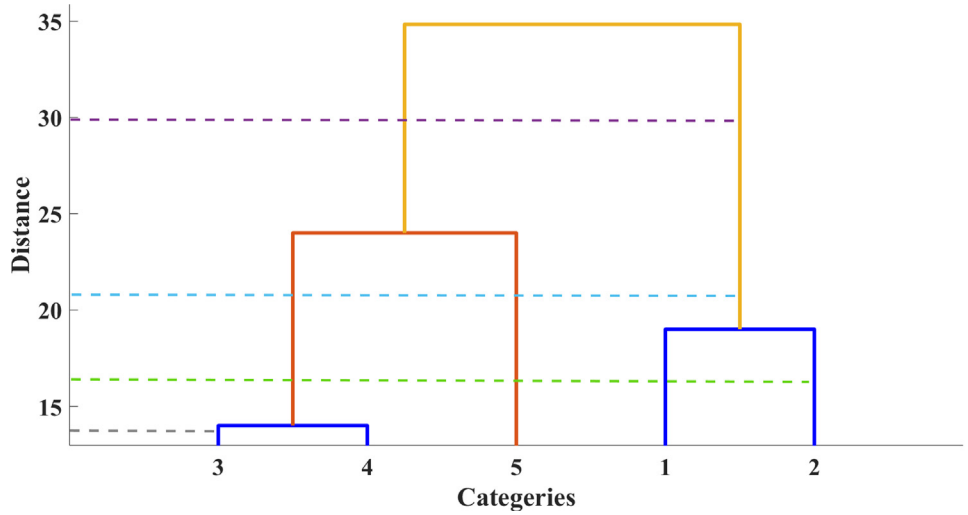

Fig. 5. Clustering tree that clusters multiple classes "1, 2, 3, 4, 5" at the bottom into one class at the top based on Levenshtein distance matrix. The time series data is the same with the data in previous figure, and every class contain five characters (consists of a, b, c, d, e) and every character contains five samples.

results $O_{prediction}$ are obtained through a fully connected structure (FC). At the same time, the previous clustering results C_r are used as classification criteria. The classification probability C_{prob} can be obtained through a softmax layer, and the category M with the largest probability $C_{prob_{max}}$ is the final classification result.

4. Case study

The development of the size and of the time scale of chemical engineering plants has led to a substantial increase in demand for energy consumption. Additionally, the operating cost associated with non-renewable energy sources such as coal, petroleum, and natural gas has been spiking in the recent decade (Silva et al., 2016). In order to meet the demand of the industrial energy requirement and reduce the reliance on non-renewable energy sources, new chemical compounds are investigated and manufactured as the energy source substitutes. Methanol has been developed as a popular source of energy, which has a high octane number and a low risk-factor. Therefore, it can be safely manufactured from a variety of carbon-based sources, generating environmentally friendly by-products and providing a cost-effective alternative to non-renewable energy sources (Wasmus and Küver, 1999). In the modern engineering industry, methanol is usually synthesized by catalytically reacting carbon monoxide, carbon dioxide, and hydrogen, which is governed by the following reactions:

$$CO + 2H_2 \rightarrow CH_3OH \tag{9}$$

$$CO_2 + 3H_2 \rightarrow CH_3OH + H_2O \tag{10}$$

$$CO_2 + H_2 \rightarrow CO + H_2O \tag{11}$$

A typical methanol production industrial processing includes the processing of the feed gas, the purification in advance to the main reactor, the synthesis of methanol, and the rectification of methanol. In this work, we will take the production plant of Hainan Petrochemical Co., Ltd. as an example, where the core process diagram is shown in Fig. 7. First, the fresh feed gas, driven by a compressor system, is mixed with the overhead recycle gas of the methanol separator. The combined synthesis gas is pressurized by a recycle gas

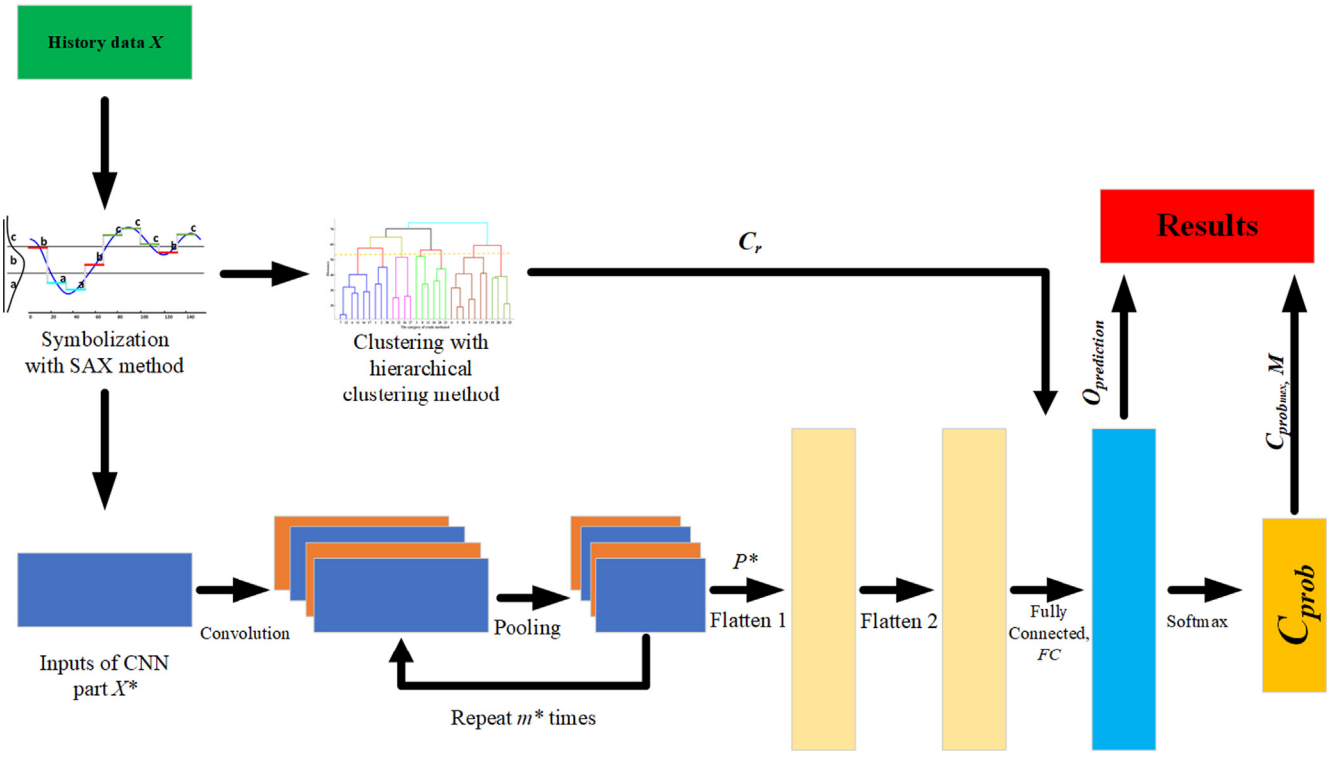

Fig. 6. Flowchart of the proposed SHC-CNN method, where the historical data is first symbolized into letters, and fed into a CNN to predict the operation trend based on clustering results.

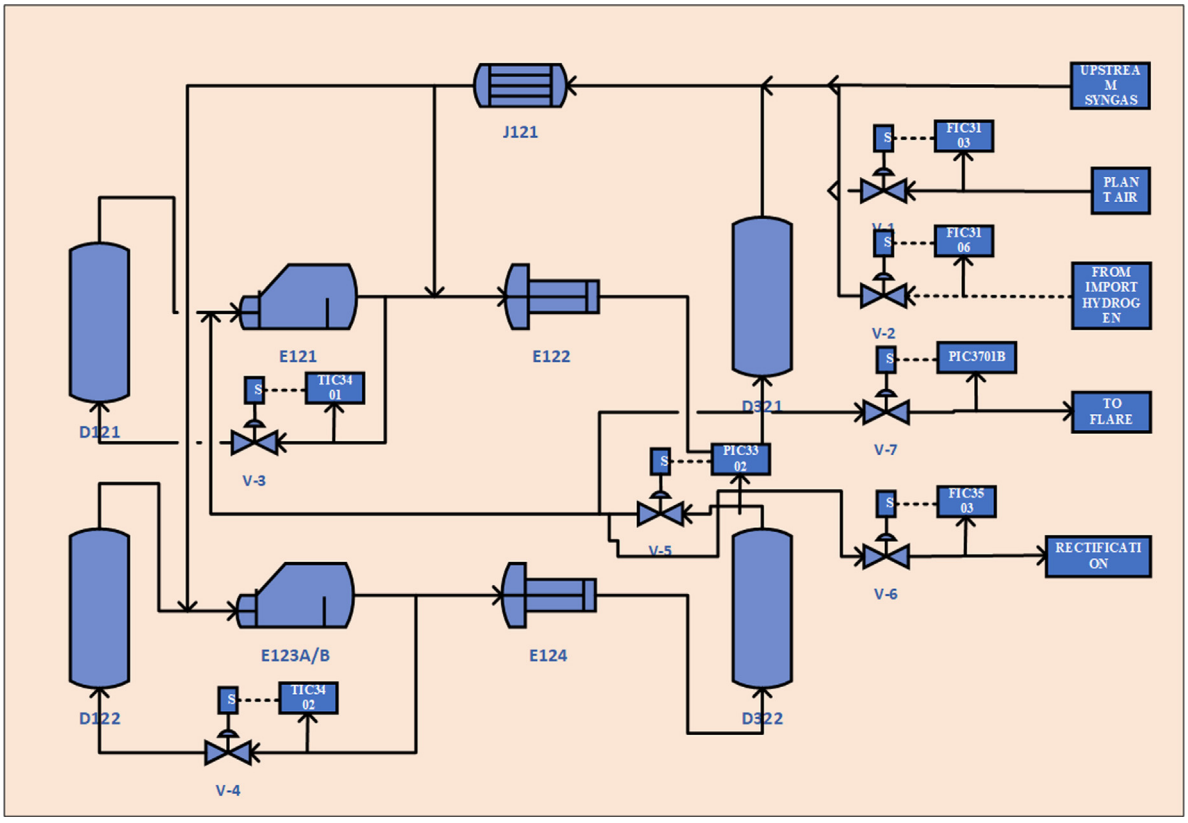

Fig. 7. Central processing part of the Hainan Petrochemical Co., Ltd. methanol production process.

Table 2
Operational variables in the central processing part of the Hainan Petrochemical Co., Ltd. methanol production process in Fig. 7.

Device Number	Variable Name	Variable Unit
FIC3103	The airflow of the factory	Nm^3/h
FIC3106	Hydrogen flow	Nm^3/h
FIC3503	Crude methanol flow	t/h
PIC3302	Outlet pressure of methanol separator	MPa
PIC3701B	Outlet pressure of methanol separator	MPa
TIC3010	The temperature of the first synthetic tower	°C
TIC3210	The temperature of the second synthetic tower	°C
TIC3401	Outlet gas temperature of the first synthetic tower	°C
TIC3402	Outlet gas temperature of the second synthetic tower	°C

compressor and preheated in a heat exchanger before entering the methanol synthesis column. Next, the hot product gas stream leaving the methanol synthesis column is cooled by a second heat exchanger and then by a water cooler. Finally, the crude methanol is then distilled off in the separator. Fig. 7 shows the most important part of the methanol production process, in which J121 represents the compressor, E121 and E122 represent the preheaters, D121 and D122 represent the synthesis reactors, E122 and E124 represent the cooling device, and D321 and D322 represent the separator.

A complex industrial process as described above requires the in-depth knowledge and extensive experience from the operators for optimal plant operation, which is necessary to ensure the efficient and safe operation of methanol production. As a result, we adopt the proposed trend prediction and classification model to assist the operator's handling of this methanol production process. In this section, the application and the results of the SHC-CNN model for the Hainan Petrochemical Co., Ltd methanol production process will be discussed in detail.

To construct a computationally tractable model that generates accurate predictions, we first need to learn about the key knowledge of the methanol production process before modeling to select variables of the most importance for the methanol production process. The list of operating parameters is shown in Table 2, where the device numbers are acquired from the actual chemical plant. For each of these operating units, a randomly chosen set of the historical operating data is shown in Fig. 8 to illustrate the typical behavior in the historical operations. From this illustration, it is demonstrated that the values of variable TIC3010 and variable TIC3210, which represent the temperature of the two synthetic towers, respectively, are always maintained at the same temperature. Therefore, these two variables replaced by the outlet gas temperature variables of the two synthetic tower, TIC3401 and TIC3402 in the input–output modeling. As a result, the crude methanol flow, FIC3503, is regarded as the final output variable, and the remaining operand variables are used as input, for the constructed SHC-CNN model.

The data is collected at a sampling time of 30 s. 20000 sets of data series are selected, in which 12000 groups of selected data are randomly chosen as the training set and 8000 groups of selected data are then used as the test set. After the historical data set is obtained, we adopt the proposed symbolization method to convert the raw numerical data series into sets of strings, which are then processed through the clustering algorithm. Because the clustering method is computationally inefficient for a large database, we divided 12000 training samples equally into 12 groups for parallel computing to improve the computational speed. In order to make it more convenient for the operator to observe the operation effect and to carry out the SHC-CNN algorithm, the same SAX clustering analysis is also performed on the training output, the crude methanol production.

Fig. 9 shows the symbolized data series corresponding to the sample numerical operational variable data, every five-second data points are characterized by a lowercase letter, and every 5 lowercase letters are processed into one category. According to the distinct characteristics observed from each of the parameter data series, the symbolized historical data are automatically clustered with different patterns into their respective types, as shown in Fig. 10. In addition, Figs. 11 and 12 show a part of the symbolized output results and a part of the hierarchical clustering results of the output crude methanol production history, respectively. It is demonstrated from Figs. 9 and 11 that the historical inputs and output numerical data are successfully replaced by different letter series. Also, as shown in Figs. 10 and 12, the symbolized data are gathered into different categories through the hierarchical clustering method. In order to reduce the computational burden, we use

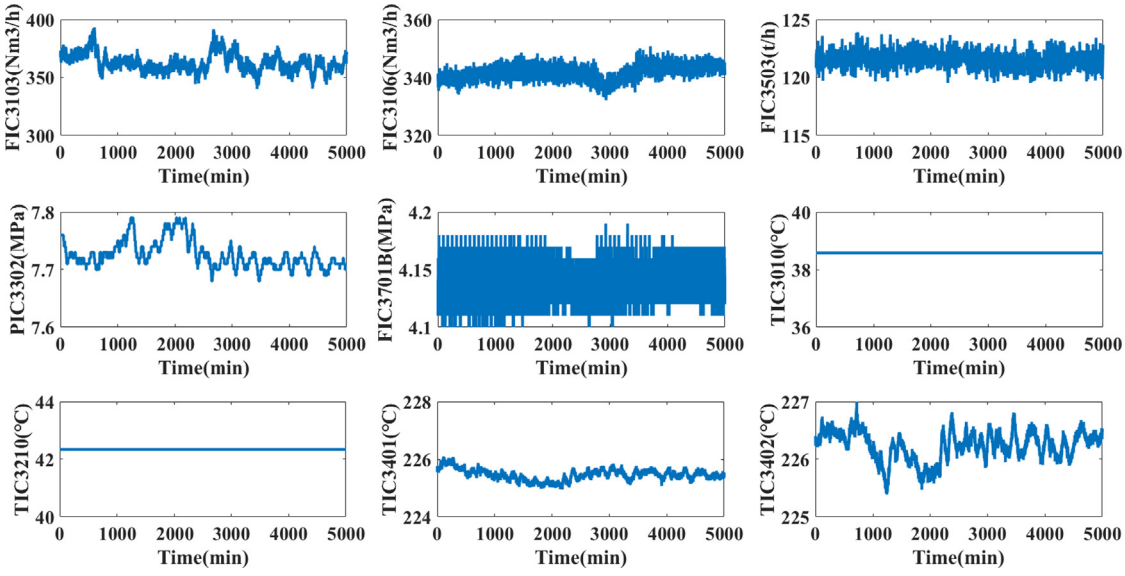

Fig. 8. Example of historical data of the selected operational variables.

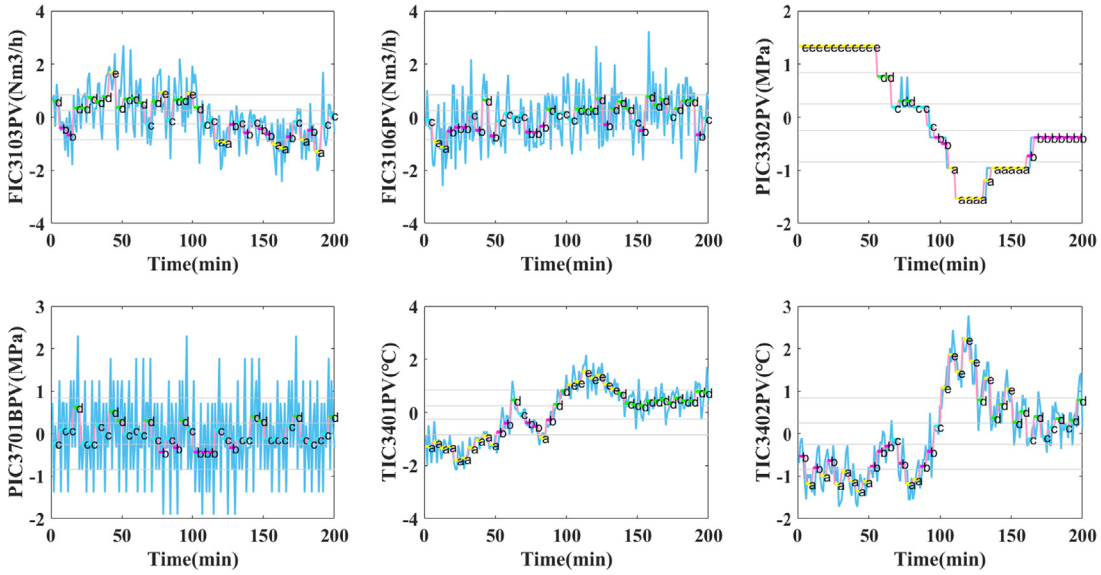

Fig. 9. Symbolization results (red profiles with the letters "a, b, c, d, e") of the operational variables based on the historical data (blue profiles). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the yellow dotted line as the boundary, and the clustering result of crude methanol is divided into seven categories based on the yellow dotted line.

Next, we input the post-processed results of symbolic clustering into CNN for classification. This model consists of 6 variables as model inputs and the time lag of the variables is one sampling point time interval. The number of levels and parameters is determined through a grid search, which ends up giving the optimal training result with 5 convolution layers, 36 filters and 3 pooling layers. There are 12 filters between the first convolution layer and the input data. The weight matrix w for every filter is of dimension 6×6, and the width is equal to the number of the input variables. The number of the bias b in every filter is the same with the number of the weights. Each of the first three convolutional layers is followed by a pooling layer, and the last two convolutional layers are the fully connected convolutional layers. It is noted that the number of the filters needs to be carefully chosen. Specifically, if the number of filters is small, we cannot extract enough features from the history data. However, a large number of filters could lead to over fitting and computation burdens. Besides, too much filters and layers also increase the computation burden of the algorithm. In addition, we use stochastic gradient descent method to train the model, where the iteration number is 200, the learning rate is 0.001, and the weight decay term is 0.0001. These values are chosen based on empirical neural network training knowledge and also via trial-and-error tuning. Also, the proposed SHC-CNN inherits the sparse connection and value sharing characteristics of traditional CNN, as described in sections "sparse connection" and "weight sharing", and thus, the proposed SHC-CNN can reduce the computational burden and improve the computational efficiency.

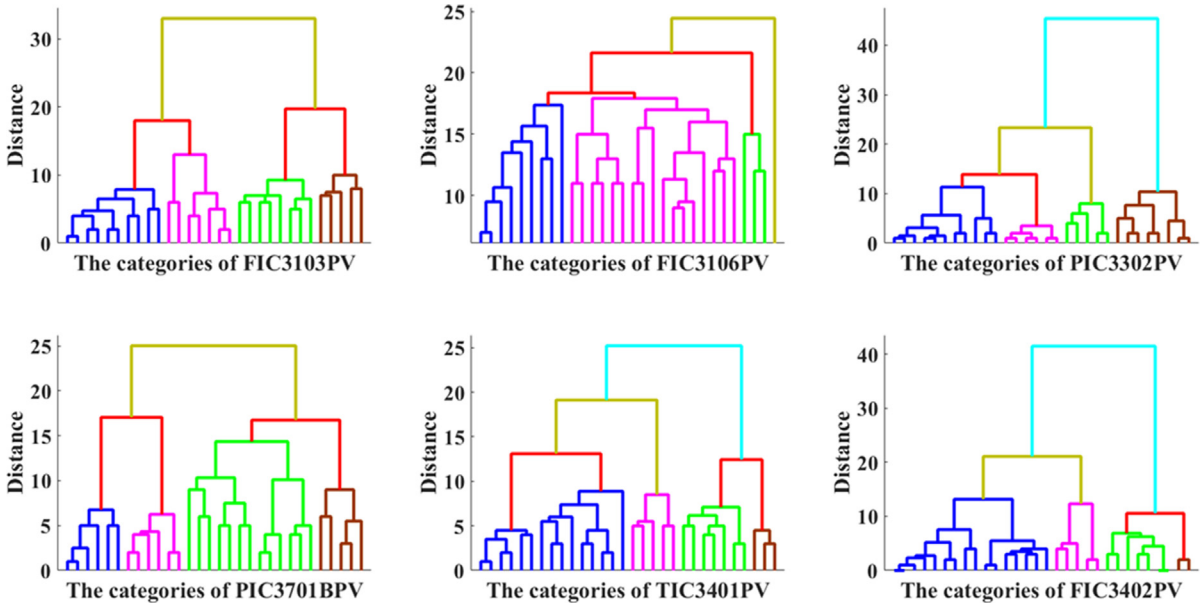

Fig. 10. Hierarchical clustering results for input operational variables.

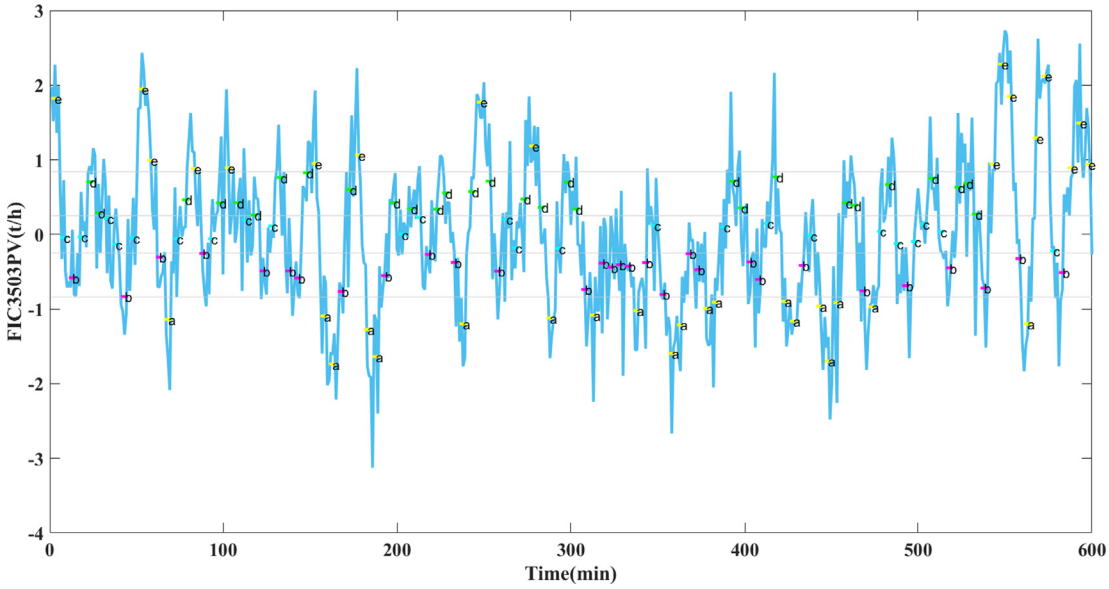

Fig. 11. Symbolization results of the historical data of crude methanol production.

The training and predicting results are shown in Figs. 13 and 14. It is demonstrated from the figures that the operation trends have been accurately predicted. a, b, c, d, e represent the five different categories, respectively, and the breakpoints can be found in Table 1. In addition, the parameters of the BP and RNN methods are both selected via grid search for optimal results. FIC3103PV, PIC3302PV, and TIC3401PV receive better results than the rest of the variables, because these three variables change more gently. We also give a trend forecast for crude methanol production that is closely related to the amount of operation, where the classification results of the test data are analyzed, as shown in Fig. 16. The classification accuracy of FIC3103PV, FIC3106PV, PIC3302PV, PIC3701BPV, TIC3401PV, TIC3402PV and FIC3503PV are 0.865, 0.846, 0.965, 0.897, 0.855, 0.903 and 0.893, respectively. Compared with historical data, the trend forecast results are highly reliable. Nevertheless, we can see from Figs. 13 and 14 that there is a time lag in our prediction result. Since we are dealing with time-series data, which tend to be correlated in time and exhibit a significant auto-correlation, the prediction results often exhibit a time lag between predicted value and true value when using common error metrics such as *RMSE* or R^2 score to evaluate model accuracy. A potential approach to addressing this issue is to use the difference between consecutive time steps, which could provide a stronger test for model accuracy in this case. In addition, it should be considered when used for other modeling that the choice of the number of key parameters will impact the optimal model accuracy but the exact correlation cannot be explicitly concluded because of the competition between the training accuracy and over-fitting.

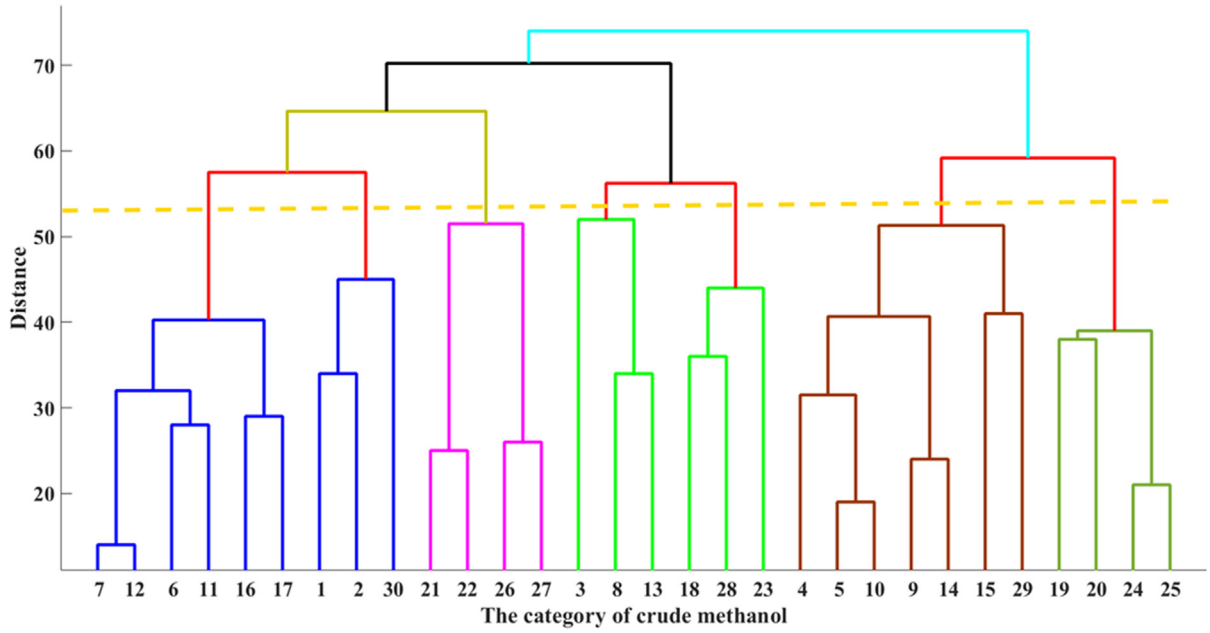

Fig. 12. Hierarchical clustering results for historical data of output crude methanol production.

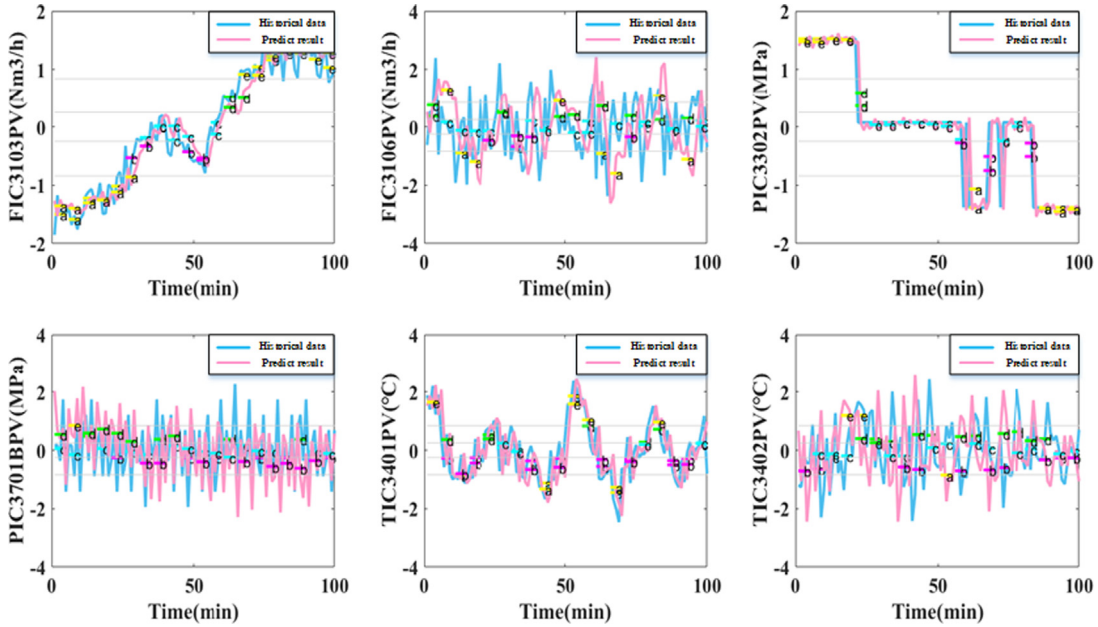

Fig. 13. Operation trend prediction results, where the blue and the red profiles represent the historical data and predicted results, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

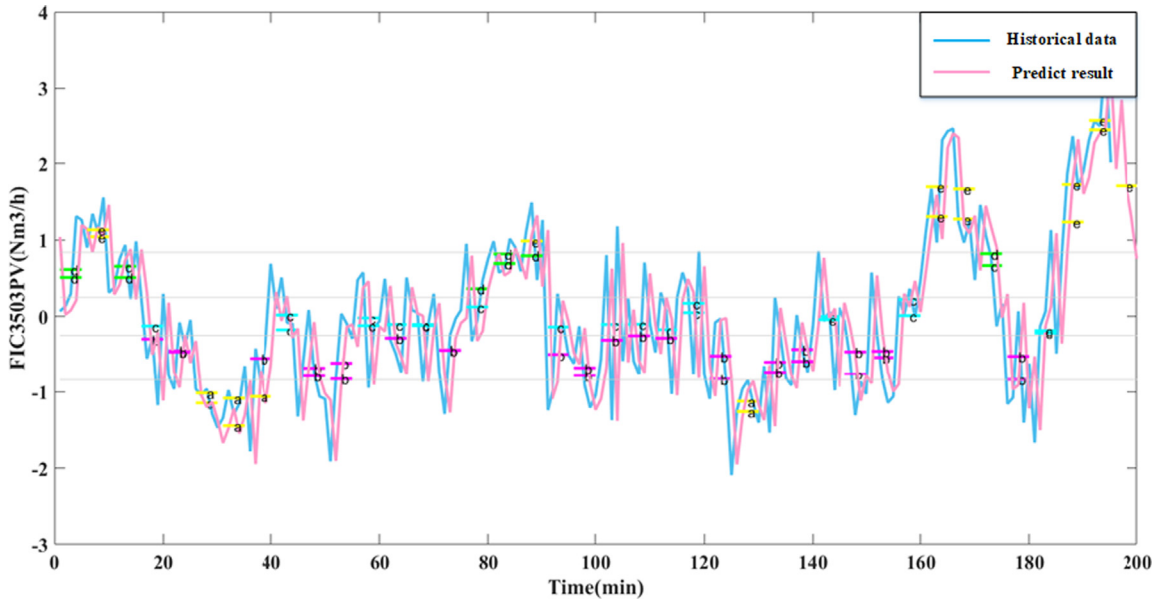

Fig. 14. Trend prediction of crude methanol production, where the blue and the red profiles represent the historical data and predicted results, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

However, the proposed structure seems to be a good starting point and a subsequent grid search can be done according to the input and output dataset dimension.

To further demonstrate the effectiveness of the proposed method, the process data is used to train a traditional CNN, an RNN and a BP neural network (BPNN) for process fitting and prediction, respectively. The traditional CNN has the exact same structure without the SHC pre-processing. As for the BP neural network, we also have the 6 variables as inputs. Three layers are used, including one input layer, one hidden layer, and one output layer, where the number of neurons in the hidden layer is selected to be 10. Similarly, in the RNN structure, there are also 3 layers: the input layer, the hidden layer and the output layer; the number of neurons in the each layer are the same as that in the BP neural network. The number of neurons and hidden layers are again determined from grid search. The testing and training of three networks in comparison all use the same training and testing dataset, respectively. The *RMSE* of traditional CNN, RNN, BPNN, and the proposed SHC-CNN is 0.3192, 0.6998, 0.8272, and 0.2917, respectively. A randomly selected set of data sample is plotted to compare the four models, which is shown in Fig. 15. It can be seen that the SHC-CNN method provides more accurate trend analysis than other prediction methods, while the BP feedforward neural network produces the worst prediction results. Additionally, the SHC-CNN network has demonstrated its superiority of predicting not only the next sampling step but also the future trends.

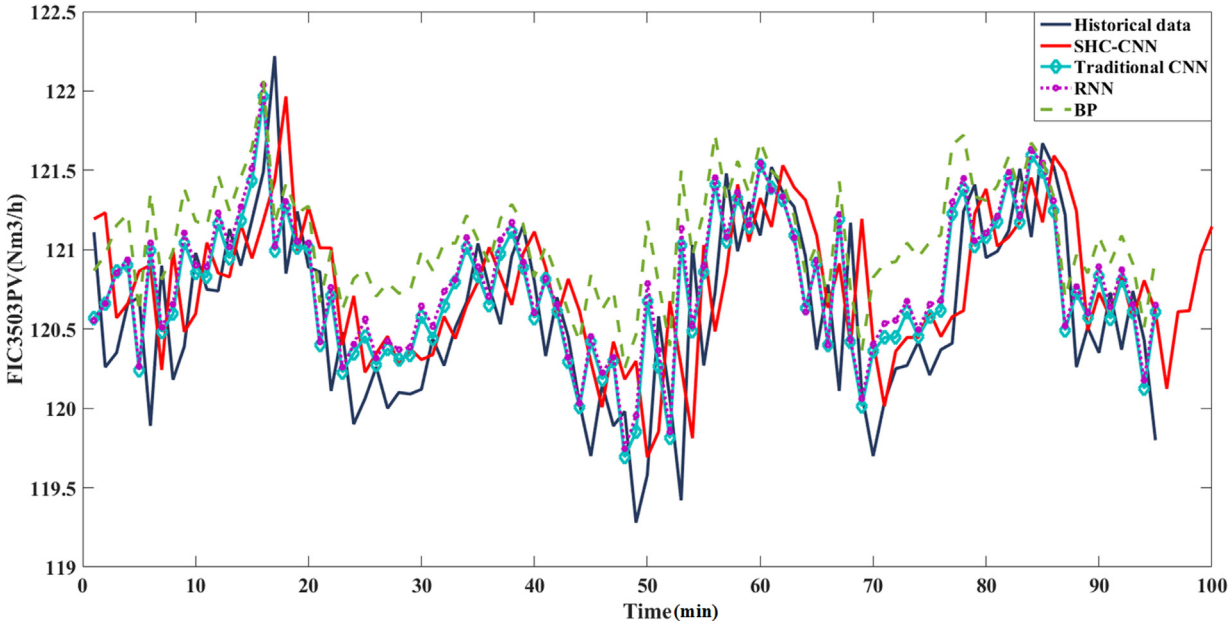

Fig. 15. Comparison of the crude methanol production from historical data, and the results predicted by the SHC-CNN, traditional CNN, RNN, and BP methods, respectively.

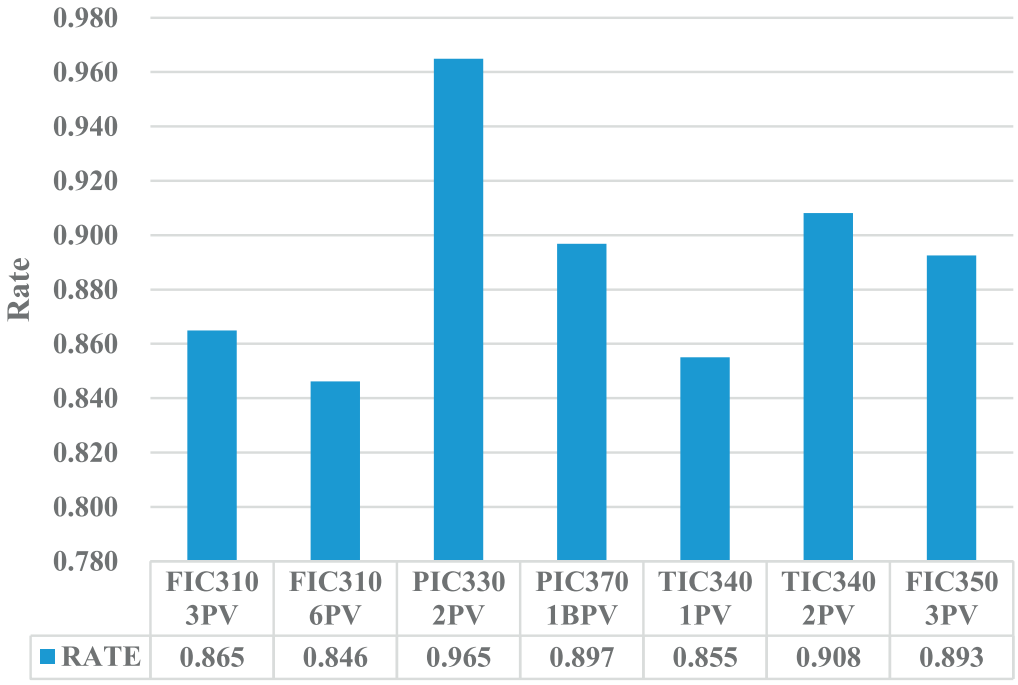

Fig. 16. Classification accuracy of FIC3103PV, FIC3106PV, PIC3302PV, PIC3701BPV, TIC3401PV, TIC3402PV and FIC3503PV based on testing data samples.

Moreover, the category and performance of the operation trend are accounted for, which greatly improves the auxiliary ability of the operator and the efficiency of the industrial process.

5. Conclusion

This paper proposed a novel convolutional neural network operation trend prediction and classification method based on symbolic hierarchical clustering (SHC-CNN). Firstly, the historical data was symbolized. Then the symbolized data was categorized into distinct classes through the hierarchical clustering method. Finally, a specifically tailored CNN network was trained with the historical data to obtain an accurate historical trend fitting. To demonstrate the effectiveness of the proposed SHC-CNN method, the methanol production process of Hainan Petrochemical Co., Ltd was applied, for which accurate fitting results were obtained for all input variables and output methanol production. Additionally, the proposed SHC-CNN algorithm was also compared with the traditional CNN and RNN, from which the superiority of the proposed trend prediction was demonstrated.The proposed SHC-CNN was demonstrated to be able to extract deep and rich features through multi-angle description of every local section. It allowed the time series data to be filtered with interference and to be

considered in a more complete picture. Overall, CNN-based methods can greatly overcome the high dependence of modeling on first-principles model. The precise prediction results from the SHC-CNN algorithm could help process engineers analyze the current state of the process, and provide reliable insights to help engineers operate the system.

CRediT authorship contribution statement

Yongjian Wang: Conceptualization, Methodology, Software, Writing - original draft. **Yichi Zhang:** Writing - original draft. **Zhe Wu:** Writing - original draft. **Hongguang Li:** Supervision. **Panagiotis D. Christofides:** Supervision, Writing - review & editing.

Declaration of Competing Interest

None.

References

Amasyali, K., El-Gohary, N.M., 2018. A review of data-driven building energy consumption prediction studies. Renew. Sustain. Energy Rev. 81, 1192–1205.

Boguslavskii, L., Kirsanov, G., 1989. Enhancement of the accuracy and sensitivity of operation of measuring devices in controlling the process of longitudinal grinding of long shafts. Meas. Tech. 32, 508–509.

Bryant, A., Cios, K., 2018. RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates. IEEE Trans. Knowl. Data Eng. 30, 1109–1121.

Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., Mathieu, C., 2019. Hierarchical clustering: objective functions and algorithms. J. ACM (JACM) 66, 26.

Ding, S., Jia, H., Du, M., Xue, Y., 2018. A semi-supervised approximate spectral clustering algorithm based on HMRF model. Inf. Sci. 429, 215–228.

Ding, Y., Zhang, Y., Ren, Y.M., Orkoulas, G., Christofides, P.D., 2019. Machine learning-based modeling and operation for ALD of SiO2 thin-films using data from a multiscale CFD simulation. Chem. Eng. Res. Des. 151, 131–145.

Dunjko, V., Briegel, H.J., 2018. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep. Prog. Phys. 81, 074001.

Iwahashi, J., Kamiya, I., Matsuoka, M., Yamazaki, D., 2018. Global terrain classification using 280 m dems: segmentation, clustering, and reclassification. Prog. Earth Planet. Sci. 5, 1.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Comput. Surv. (CSUR) 31, 264–323.

Kim, S.H., Whitt, W., Cha, W.C., 2018. A data-driven model of an appointment-generated arrival process at an outpatient clinic. INFORMS J. Comput. 30, 181–199.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inform. Process. Syst., 1097–1105

LeCun, Y., et al., 2015. Lenet-5, convolutional neural networks. URL: http://yann.lecun. com/exdb/lenet 20, 5.

Li, X., Wang, N., Wang, L., Kantor, I., Robineau, J.L., Yang, Y., Maréchal, F., 2018. A data-driven model for the air-cooling condenser of thermal power plants based on data reconciliation and support vector regression. Appl. Therm. Eng. 129, 1496–1507.

Lu, J., Zhu, Q., Wu, Q., 2018. A novel data clustering algorithm using heuristic rules based on k-nearest neighbors chain. Eng. Appl. Artif. Intell. 72, 213–227.

Lu, S., Lu, Z., Zhang, Y.D., 2019. Pathological brain detection based on alexnet and transfer learning. J. Comput. Sci. 30, 41–47.

Lukoševičius, M., Jaeger, H., 2009. Reservoir computing approaches to recurrent neural network training. Comput. Sci. Rev. 3, 127–149.

Mohamed, A.r., Dahl, G.E., Hinton, G., 2011. Acoustic modeling using deep belief networks. IEEE Trans. Audio, Speech, Lang. Process. 20, 14–22.

Murtagh, F., 1983. A survey of recent advances in hierarchical clustering algorithms. Comput. J. 26, 354–359.

Peterson, A.D., Ghosh, A.P., Maitra, R., 2018. Merging k-means with hierarchical clustering for identifying general-shaped groups. Stat 7, e172.

da Silva, R.C., de Marchi Neto, I., Seifert, S.S., 2016. Electricity supply security and the future role of renewable energy sources in brazil. Renew. Sustain. Energy Rev. 59, 328–341.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Smarra, F., Jain, A., De Rubeis, T., Ambrosini, D., D'Innocenzo, A., Mangharam, R., 2018. Data-driven model predictive control using random forests for building energy optimization and climate control. Appl. Energy 226, 1252–1272.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.

Tong, Y., Zhong, W., Tong, S., 2015. Performance online prediction of supercritical boilers based on mechanism and data-driven model. Proc. CSEE 35, 2487–2494.

Wang, Y., Li, H., 2018. A novel intelligent modeling framework integrating convolutional neural network with an adaptive time-series window and its application to industrial process operational optimization. Chemomet. Intell. Lab. Syst. 179, 64–72.

Wang, Y., Li, H., Huang, J., Su, C., Yang, B., Qi, C., 2019. An improved bar-shaped sliding window cnn tailored to industrial process historical data with applications in chemical operational optimizations. Industr. Eng. Chem. Res. 58, 21219–21232.

Wasmus, S., Küver, A., 1999. Methanol oxidation and direct methanol fuel cells: a selective review. J. Electroanal. Chem. 461, 14–31.

Xu, X., Ding, S., Du, M., Xue, Y., 2018. DPCG: an efficient density peaks clustering algorithm based on grid. Int. J. Mach. Learn. Cybernet. 9, 743–754.

Yu, X., Zhang, X., Qin, H., 2018. A data-driven model based on fourier transform and support vector regression for monthly reservoir inflow forecasting. J. Hydroenviron. Res. 18, 12–24.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2019. Deep learning and its applications to machine health monitoring. Mech. Syst. Signal Process. 115, 213–237.