2020 IEEE 9th Data Driven Control and Learning Systems Conference
November 20-22, 2020, Liuzhou, China

A Multi-Channel Multi-Head CNN Framework for Fault Classification in Industrial Process

Hui Wu[1], Yan Wang[2], Junyu Lin[1], Weidong Yang[1], Yanwei Wang*[3], Ying Zheng[1]

1. The Key Laboratory of Image Information Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

2. School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

3. School of Mechanical Electrical Engineering, Wuhan Institute of Technology, Wuhan 430073, China
E-mail: ywwang.cad@gmail.com

Abstract: This paper proposes a novel fault classification method via convolutional neural network with multi-channel and multi-head along the time dimension, which is defined as MM-CNN. The MM-CNN extracts features of industrial process data by convolutional layers from local to global level. Unlike traditional methods, this method can capture the independent features of every process variable and the relevant characteristics from sensor data to learn more useful latent fault patterns. Besides, a data preprocessing approach is proposed to transform original data for convolutional neural network. Finally, for all the all 21 faults in Tennessee Eastman(TE) process, this paper compares the proposed method with the published methods and a architecture which combines the convolutions of multi-head and a bidirectional Long Short-Term Memory (Bi-LSTM) layer in fault classification. The simulation results show that the method has better classification performance than the state-of-the-art methods.

Key Words: deep learning, one-dimensional convolutional neural network, fault classification, feature extraction

1 Introduction

As the model industrial systems becoming increasingly complex and integrated, process monitoring of industrial systems becomes more and more important. A simple fault might damage the functional portions, thus further to degrade the overall system resulting in economic losses and even huge casualties. Reducing harm, improving the safety and reliability of equipment manipulations, and decreasing manufacturing costs, timely and precise fault detection and fault diagnosis for the real process are vital. In general, process monitoring methods can be divided into three types: model-based methods, knowledge-based methods and data-driven methods [1, 2]. In practice, it is difficult to obtain the physical principles and prior knowledge of the systems. With the rapid developments in storage technology, numerous important measurements, and final product quality variables are collected and stored by on-line measurements and off-line analyses, as a result, data-driven process monitoring technologies can be ever-accelerated and become more and more popular [3]. In recent years, Deep-Learning-data-based fault classification methods have been developed, and become the trend of fault diagnosis.

Several data-driven fault classification techniques have been proposed and greatly improved over the past few years. There are mainly two types of data-driven fault classification techniques: linear supervised classification techniques and non-linear classification techniques. Linear supervised classification techniques: Rule-based classifier, Nearest-Neighbor classifier [4], Bayesian classifier, Principal Component Analysis (PCA), Decision Trees, Fisher discriminant analysis (FDA) [5], Partial Least Squares (PLS); Non-linear classification techniques: Artificial Neural Networks (ANN) , Support Vector Machine (SVM) [2] and Deep Neural Network (DNN) [2, 4–7]. Among them, SVM, PCA and DNN are widely used in fault classification in recent years. DNNs have indeed revolutionized the field of computer vision [8], sequential data such as text and audio[9]. Among these methods, recent works have considered Deep Neural Networks to perform this task are on the rise. Supervised and semi-supervised learning strategies have been introduced for fault classification and diagnosis in industrial processes. Many deep learning models will be combined with the signal analysis method, which can extract features of processed signals. Then the features are delivered to the trained classification model and taken as the feature for the fault classification. S-transform, wavelet transform, and fast Fourier transform (FFT) are adopted to analyze the fault signals. It is usually taken as the feature for the fault classification. For example, [10] proposes a fault detection and classification method of the deep residual convolutional neural network. It captures the deep process features represented by convolutional layers from local to global and can extract the deep fault information and learn the latent fault patterns. [11] used Hilbert-Huang transform and convolutional neural for fault classification. [3] developed an ensemble form of the semi-supervised Fisher Discriminant Analysis (FDA) mode for fault classification in industrial processes with the limited number of labeled faulty samples.

To resolve the problems mentioned above, this paper proposed a novel end-to-end method for fault classification in the industrial process. The main contribution of this article is as follows: First, we proposed a novel one-dimensional convolutional neural network for fault classification, which extracts the independent features of every process variable and the relevant characteristics from sensor data, and obtained a state-of-the-art performance in Tennessee Eastman fault classification of all 21 faults. Second, our proposed architecture is more simple than the existing methods of deep learning methods, so it can classify faults online. Third, we compare our proposed architecture with a model that combines MM-1D-CNN and a bidirectional Long Short-Term Memory (Bi-LSTM) [12] layer to prove the method in this

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61873102.

paper is powerful for 21 faults classification of TE process.

The rest of this paper is organized as follows. Section 2 describes our proposed MM-CNN network. Section 3 reports the evaluation results of the proposed model in comparison with strong baselines on TE fault data sets. Finally, the conclusions are presented in Section 4.

2 The Proposed Framework

The general overview of the MM-CNN architecture is present in Fig. 1. The configuration of the architectures for extracting features is divided into two groups: Multi-head and Multi-channel. The Multi-channel CNN (MC-CNN) uses a single convolutional head with multiple channels to process the time series. The Multi-head CNN (MH-CNN) uses independent single-channel convolutional heads to process each Process Variable separately. This architecture combines MC-CNN and MH-CNN, which is just simply convolution along the time dimension. Firstly, MC-CNN and MH-CNN are both responsible for extracting meaningful features from sensor data separately. For the MC-CNN, a 2D average pooling is applied over the extracted features of several input planes, then the outputs of the 2D average pooling are as the inputs of the Group Normalization (GN) layer [13]. For the MH-CNN, it extracts the features of each time series independently, then a 1D average pooling and a Group Normalization (GN) layer are followed. Secondly, in this paper, features extracted from MM-CNN are connected in series, and then ReLU [8] activation layer is applied. Next, a dropout layer is used to regularize the activation to avoid overfitting. Finally, a dense layer follows by the Batch Normalization (BN) layer. Next, the SoftMax layer is applied to generate the output of the architecture. The inclusion of the BN layer is particularly important as it reduces the internal covariance shift. This brings a regularization effect between batches and makes training faster.

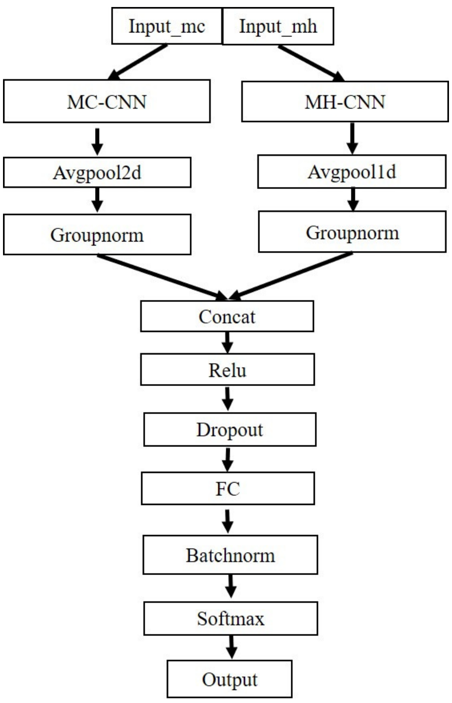


Fig. 1: MM-CNN architecture.

2.1 Convolutional Component

In deep learning model, since AlexNet [8] won the ImageNet competition in 2012, convolutional neural networks (CNN) have been used in many different domains such as image recognition, natural language processing and time series analysis. Convolution can be seen as applying and sliding a filter over sequence data. In an industrial process, data of the process variables can be seen as an image to be processed by CNNs. Also, the characteristics of data in the industrial process should be considered. For process variables, they are time series and some variables are correlated or independent [14]. Unlike other approaches, the filters exhibit only one dimension (time) instead of two dimensions(width and height). Concretely, if a filter of length 3 is convolved (multiplied) with a univariate time series, by setting the filter values to be equal to [1/3,1/3,1/3], the convolution will result in the application of the a moving average with a sliding window of length 3. A general form of applying the convolution for a centered time stamp t is given in the following Fig. 2. A convolution can be defined as follow:

$$\mathrm{y} = \sum_{i=1}^{M}\sum_{j=1}^{N} x(i,j)w(i,j) \quad (1)$$

where y is the output of convolution from input x(i,j) and a filter w(i,j) with the length and the width of M and N respectively.

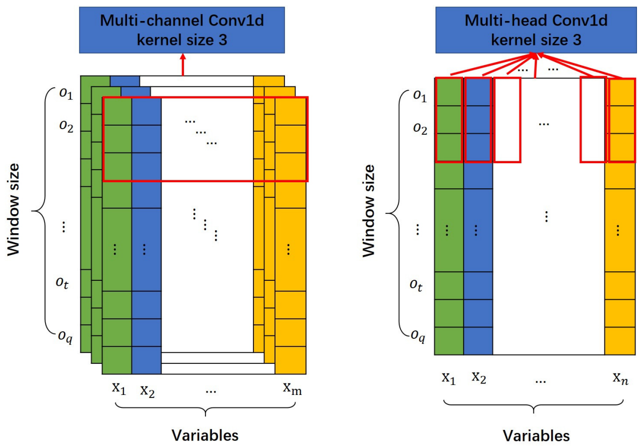


Fig. 2: Architecture of Multi-channel CNN (left) and Architecture of Multi-head CNN (right).

The aim of the convolutional part of the proposed architecture is to extract the most characteristics from sensor data. The convolution can be seen as applying and sliding a filter over process data. In this paper, when dealing with multiple time series, CNNs with multiple channels are used where each channel corresponds to one type of variables, which is called Multi-channel CNN. Its general form of applying the convolution for a centered time stamp t is shown in left of Fig. 2. When a Multi-channel CNN is used to process multiple time series, a feature map containing the main features of all the time series is obtained as a result. In contrast, the Multi-head CNN extracts the features of each time series independently, its form of applying the convolution for a centered time stamp t is shown in Fig. 2. As a consequence, an independent feature map for each time series is obtained. So, this proposed method can capture useful features from local to global level.

The typical loss function is cross-entropy.Let N be the number of samples, the loss function of cross-entropy can be calculated as:

$$\text{loss} = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right] \quad (2)$$

where y_n is the probability of true label, $\hat{y}_n$ is the probability of predicted label.

3 Case Study

Tennessee Eastman (TE) process, developed by Eastman Chemical Company, is a real chemical process for process control and monitoring evaluation. The TE process is applied to evaluate the methods proposed in Section 2. TE process, first introduced by Downs and Vogel, has been widely used for testing and evaluating various process monitoring algorithms and control strategies. 21 faults data are also available for simulation in this process, whose detailed descriptions are listed in Table 1, which include sixteen known faults and five unknown The process can also simulate 21 fault data, including 16 known faults and 5 unknown faults. More detailed description of the TE process can be found in reference [15].

The data in the TE simulation dataset come from 22 different simulation data samples. One of the simulations occurs under the no-fault mode, and each of the rest simulations happens under different fault modes. Every fault has a training set with 480 samples and a test set with 960 samples. TE process has been widely adopted as an experiment of a continuous process to evaluate the performance for fault detection methods [2, 4–7]. There are 52 variables including manipulated variables and measured variables in TEP. In measured variables, there are process variables and component variables. 21 identified faults for different types are listed in detail in Tables **??**. The train set is sampled once every three minutes and the process simulation time is 25 h, which faults began in the first beginning of the second hours, that is to say, faults are introduced ever since the 21th data. The test set is sampled once every three minutes and the process simulation time is 48 h, which faults are introduced ever since the 161th data. So, In the training set, the fault is generated from sample 21 to the end. In the test set, the fault is generated from sample 161 to the end. To prepare data for CNNs, supposing the input data is $X = \{O_1, O_2, \cdots, O_T\}$, T is the total numbers of sampling data. A tunable window with size q in time dimension is used to reformulate the input at timestamp t as $X_t = \{O_{t-q+1}, O_{t-q+2}, \cdots, O_t\}$. Then series set of feature-target pairs$\{X_t, Y_t\}$ would be gained, where Y_t is the fault type of X_t, $t \in [q, T]$. For MC-CNN, there are 11 manipulated variables, 22 process variables and 19 component variables are organized as input data of every channel, which are all padding zeros to 22 dimensions. For MH-CNN, it just process all variables in time dimension.

The Fig. 3 is the visualization of the raw data of TE fault datasets through t-SNE [16] in two dimensions. From the visual picture, many different fault types are aliasing together.

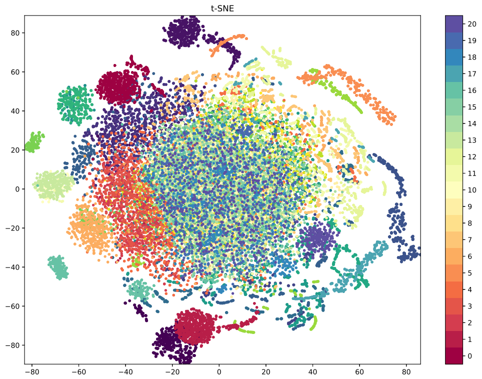


Fig. 3: The raw data of TE fault datasets are projected into two dimensions.

There have some methods for TE fault classification [2, 4–7], but most of them generalize badly on the test set. It is a very challenging task to classify all fault types properly. In the published methods for classifying fault types of TE process is usually just to select several faults in the all 21 faults, especially, most of the deep learning methods removed some faults to train a classifier [4, 6, 7], this paper designs a classifier for all 21 faults.

3.1 Methods for Comparison

We used a comprehensive set of fault classification methods to compare the performance of our method. K-Nearest Neighbors [4], c-SVM [2], FDA [5] stands the state-of-the-art baselines for fault classification. LightGBM [17] is a highly efficient gradient boosting decision tree, which is a popular ensemble learning algorithm. It shows great power in Classification and regression. The hyper-parameters of these methods are obtained by conducting 10-fold cross-validation on the test set for minimal validation error.

The MC-CNN only uses multi-channels CNN to extract features. The MH-CNN extracts features by multi-head CNN. The other parts of MC-CNN and MH-CNN are both unchanged. The deep learning models use the same hyper-parameters and the comparable number of parameters.

Another popular type of architectures for deep learning models is the Recurrent Neural Network (RNN). CNN-LSTM is an architecture that combines convolutional and recurrent layers. The output of the multi-head convolutional layer is fed into the Bidirectional LSTM (Bi-LSTM) which can access long-range time series in both input directions. Next, a dropout layer is used to regularize the activations to avoid overfitting. Finally, a dense layer is applied to generate the output of the architecture. For this layer, a SoftMax activation function is used whose output is a probability distribution. This probability distribution refers to the fault type of input, thus it is then rounded to obtain a classification result. For deep leraning methods, our optimization strategy is a widely used back-propagation algorithm, Adam [18] and Rmsprop [19], to solve a classification task.

In this paper, the performance of these classifiers can be measured in terms of accuracy, precision and recall. These metrics are calculated from a confusion matrix, which displays the crossing correct and wrong predictions between pairs of categories (classes). The accuracy is the corrected classified samples percent of all test set. The precision and recall are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where TP (true positives), FP (false,positives), TN (true negatives), FN (False Negatives), which are defined in Tables 1.

Table 1: Confusion matrix.

	Number of samples in the ith class (Actual)	Number of samples in the other class (Actual)
Number of samples in the ith class (Predicted)	TP	FP
Number of samples in the other class (Predicted)	FN	TN

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

3.2 Configuration of the Models

In this section, the configuration and the hyper-parameters of the experimental model are detailed. To the end of this section, the experimental environment are executed.

All tunable hyper-parameters are obtained by conducting grid search on the test set for each method. Specifically, all methods share the same grid search range of the window size q that follows: q=16,32,64,128. The number of convolutional layers and filters were determined by trial and error so that the minimal validation error is obtained. For MH-CNN, the number of heads are 52, which are the number of variables in TEP.The configuration of MM-CNN is detailed in Tables 2. All convolutional layers use padding to maintain the previous size. Its parameters are denoted as Conv-(kernel size)-(number of filters)-(stride)-(padding)-(groups). Pool layers are denoted as Avgpool-(kernel size)-(stride)-(padding). Groupnorm layers are denoted as Groupnorm-(groups)-(channels). The Dropout layer is denoted as Dropout-(drop rate). Despite the differences in model structure, the layer configuration of other types of architecture unchanged. For CNN-LSTM, after the same MH-CNN, adding one Bi-LSTM layer whose hidden size is 256. The training epochs and the optimization strategy is adopted by their best performance.

All the experiments have been assassinated on a single computer with the following characteristics:

- GPU: 2080Ti.
- Computing platform: CUDA 10.1.
- Operating System: Ubuntu 18.04.2 LTS.

3.3 Main Results

In this section, a comparison between MM-CNN and the other methods mentioned in Section 3.2 is performed. First, the training and testing of MM-CNN, MC-CNN, MH-CNN and CNN-LSTM are described. Second, the fault classification results of all referenced above algorithms are revealed, and the performance of the different classifiers is analyzed. Finally, the difference of the 4 convolutional architecture is studied. Testing MM-CNN as the window length varies, the performance of this models, in Fig. 4, remains stable regardless of the length of the window. In this paper, the window size q = 64, the training epochs is 100, and the Adam is used to optimize our deep neural network.

Table 2: Configuration of MM-CNN.

Input	
Multi-channel Conv1d-3X22-128-1-1-1	Multi-head Conv1d-3-52-1-0-52
Avgpool2d-3X3-1-1	Avgpool1d-3-1-1
Groupnorm-128-128	Groupnorm-52-52
Concat	
Relu	
Dropout-0.1	
FC	
Batchnorm	
Softmax	
Output	

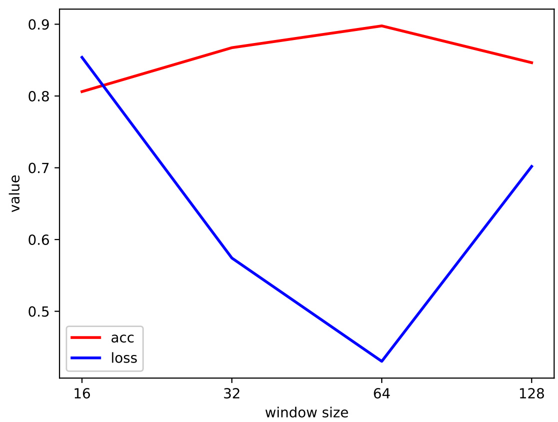
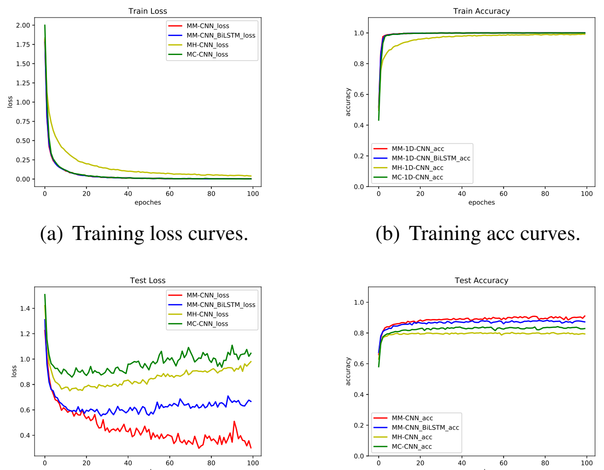


Fig. 4: Performance of the MM-CNN as the window length varies.

The training curves and test curves are shown in Fig. 5 to demonstrate the excellent performance of MM-CNN and three contrastive models in fault classification. The Fig. 5 a and b shows the values of cost functions and the accuracy in training set. The Fig. 5 c and d displays the values of cost functions and the accuracy in test set. In train set, when the steps are more than 20, costs are nearly equal to 0 and train accuracy keeps the value of 1. It means that the MM-CNN model has already learned almost all knowledge from train set. In test set, the test accuracy is lower near 20 % than train set because of the small size of train set. The performance of MM-CNN and MM-LSTM are similar, and both have better accuracy in the 21 faults classification of TE process. MM-CNN and MM-LSTM are generalized better than the other two models, which shows that the mixture of MC-CNN and MH-CNN would be obtaining stronger generalization ability.

Table 3: Classification performance for all 21 faults in TE process of different methods.

Faul Type No.	k-Nearest Neighbors	SVM	FDA	LightGBM	MC-CNN	MH-CNN	MM-CNN	CNN-LSTM
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	0.5	0.61	0.88	1	1	0.97	1	1
4	0.77	0.96	1	1	1	0.99	1	1
5	0.72	0.99	1	1	0.98	0.98	1	0.9
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	0.21	0.06	0.18	0.72	0.89	0.66	0.98	0.93
9	0.19	0.21	0.17	0.15	0.43	0.31	0.38	0.43
10	0.14	0.1	0.25	0.68	0.75	0.57	0.81	0.35
11	0	0.74	0.69	1	0.97	0.95	0.98	0.94
12	0.33	0.39	0.4	0.93	1	0.91	1	0.99
13	0.18	0.25	0.22	0.17	0.45	0.32	0.41	0.64
14	0.91	0.46	0.39	1	1	1	1	1
15	0.69	0.07	0.11	0.57	0.36	0.52	0.6	0.36
16	0.05	0.21	0.37	0.69	0.89	0.68	0.87	0.68
17	1	1	1	1	1	0.98	1	0.99
18	0.94	0.21	0.25	0.96	0.88	0.84	0.88	0.93
19	0.08	0.52	0.56	1	0.95	0.99	1	0.97
20	0.02	0.81	0.95	0.97	0.91	0.83	0.92	0.86
21	0.04	0.01	0.02	1	0.09	1	0.97	0.98



(a) Training loss curves. (b) Training acc curves.

(c) Testing loss curves. (d) Testing acc curves.

Fig. 5: Training curves and test curves for MM-CNN fault classification.

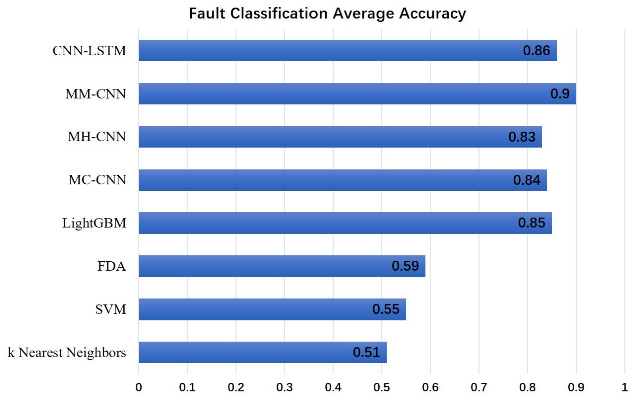


Fig. 6: Classification performance of 8 methods in test set with test average accuracy.

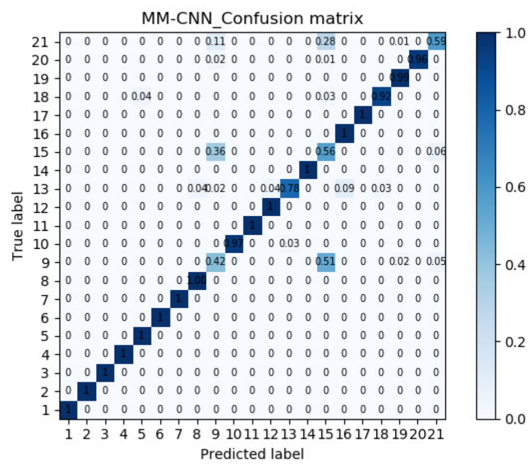


Fig. 7: Confusion matrix of MM-CNN.

The evaluation results of all the methods on the test set are summarized in Fig. 6. The MM-CNN is the state-of-the-art model. The average classification accuracy of CNN-LSTM is lower 4% than MM-CNN, but the computational complexity of CNN-LSTM is much more complex than MM-CNN because the recurrent network. The FDR of each fault type is in Tables 3. The fault diagnosis rate (FDR) represents the correct classification rate of samples with the corresponding label, which is Recall.

The confusion matrix of MM-CNN is in Fig. 7, the outstanding classification performance is displayed. Seen clearly in Tables 4, our method performs well at most fault types except 9, 13 and 15, which are also difficult for other existed methods [4, 6, 7] to classify. In 21 faults classification of TE process, the MM-CNN (our proposed) achieves superior classification performance among other methods to the best of our knowledge.

Table 4: Classification performance of MM-CNN.

Fault Type No.	Precision	Recall	F1-score
1	1	1	1
2	1	1	1
3	1	1	1
4	0.99	1	0.99
5	0.96	1	0.98
6	1	1	1
7	1	1	1
8	0.71	0.98	0.82
9	0.4	0.38	0.39
10	0.85	0.81	0.83
11	0.99	0.98	0.99
12	0.91	1	0.95
13	0.94	0.41	0.57
14	1	1	1
15	0.46	0.6	0.52
16	0.96	0.87	0.91
17	1	1	1
18	0.97	0.88	0.92
19	0.9	1	0.95
20	0.97	0.92	0.94
21	0.98	0.97	0.98

The above experimental results show that add a Bi-LSTM layer to the MM-CNN has no positive effects on TE faults classification. The distinct difference between MM-CNN and CNN-LSTM is that the CNN-LSTM performs badly at fault 10, besides fault 9, 13, 15. In LSTM layer, the long-term temporal dependencies are expected to be captured. Because in the TE process, the classification results are similar of MM-CNN and CNN-LSTM, it shows that the long-term temporal dependencies were not be learned. More research is needed to explain the reason.

4 Conclusions

In this paper, a novel architecture MM-CNN is proposed to solve the problem of fault classification in industrial process control system. MM-CNN combined with convolutional neural network with multi-channel and multi-head to extracts high-order features. These features represent the underlying correlations and latent fault patterns from the original data. By using the proposed algorithm, the vector data are transformed into matrices to train MM-CNN. The proposed method is efficient for solving complex and multivariate fault classification problems, with the feature extraction and learning abilities of MM-CNN. The experimental results on TE process show that this proposed method is superior to other methods for fault classification. As the fault classification of test accuracy for all 21 types of faults is 90%, its performance is higher 19.1% then the newest model for TE faults classification, which is called DRCNN [10]. In addition, the online test speed of the proposed method is fast, and the online test time of each sample is less than 1 ms.

References

[1] S Joe Qin. Process data analytics in the era of big data. *AIChE Journal*, 60(9):3092–3100, 2014.

[2] Chen Jing and Jian Hou. Svm and pca based fault classification approaches for complicated industrial process. *Neurocomputing*, 167:636–642, 2015.

[3] J. Zheng, H. Wang, Z. Song, and Z. Ge. Ensemble semi-supervised fisher discriminant analysis model for fault classification in industrial processes. *ISA Transactions*, 92:109–117, 2019.

[4] Q Peter He and Jin Wang. Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE transactions on semiconductor manufacturing*, 20(4):345–354, 2007.

[5] QingJun Song, HaiYan Jiang, and Jing Liu. Feature selection based on fda and f-score for multi-class classification. *Expert Systems with Applications*, 81:22–27, 2017.

[6] Yalin Wang, Zhuofu Pan, Xiaofeng Yuan, Chunhua Yang, and Weihua Gui. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA transactions*, 2019.

[7] Hao Wu and Jinsong Zhao. Deep convolutional neural network model based chemical process fault diagnosis. *Computers & Chemical Engineering*, 115:185–197, 2018.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] Xuemei Ren, Yiping Zou, and Zheng Zhang. Fault detection and classification with feature representation based on deep residual convolutional neural network. *Journal of Chemometrics*, 33(9), 2019.

[11] Mou-Fa Guo, Nien-Che Yang, and Wei-Fan Chen. Deep-learning-based fault classification using hilbert-huang transform and convolutional neural network in power distribution systems. *IEEE Sensors Journal*, 2019.

[12] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

[13] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[14] Mikel Canizo, Isaac Triguero, Angel Conde, and Enrique Onieva. Multi-head cnn–rnn for multi-time series anomaly detection: An industrial case study. *Neurocomputing*, 363:246–260, 2019.

[15] James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

[16] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2018.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] T Tieleman and G Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Tech. Rep., Technical report*, page 31, 2012.