Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Enhancing Fault diagnosis for Chemical Processes via MSCNN with Hyperparameters Optimization

Jingkang Liang[a], GürkanSin[a]*

[a] Process and Systems Engineering Center (PROSYS), Department of Chemical and Biochemical Engineering, Technical University of Denmark (DTU), 2800 Kgs.Lyngby, Denmark
* Corresponding Author: gsi@kt.dtu.dk

ABSTRACT

Fault diagnosis is critical for ensuring the safety and efficiency of chemical processes, as undetected faults can lead to catastrophic consequences. While deep learning-based methods have shown promise in this field, they often require manual hyperparameter tuning, which is not efficient since they heavily rely on expert knowledge and need iterative trial-and-error. This work introduces a novel approach combining a Multiscale Convolutional Neural Network (MSCNN) with Tree-Structured Parzen Estimator (TPE) for automated hyperparameter optimization to enhance the performance of fault diagnosis for chemical processes. The Multi-Scale Module is to capture complex nonlinear features from the fault data, while the TPE efficiently searches for optimal hyperparameters for MSCNN. An experimental study was carried out on the Tennessee Eastman Process (TE Process) dataset, where the proposed method was benchmarked against state-of-the-art models. The results indicate that the MSCNN-TPE method demonstrated improved performance in terms of precision and recall, achieving 5.26% and 5.63% higher values, respectively, compared to the CNN model. Comparisons of the MSCNN with default hyperparameters further confirmed the effectiveness of these techniques in improving fault diagnosis performance. Additionally, model ensemble technique was explored to further enhance the performance of the model and provide uncertainty estimations. In conclusion, this approach offers a robust and reliable solution for fault diagnosis in the chemical industry, enhancing process safety and efficiency.

Keywords: Fault Detection, Machine Learning, Artificial Intelligence

1. INTRODUCTION

Fault diagnosis is critical in manufacturing, especially in chemical process, to prevent failures, minimize downtime, and avoid costly repairs and hazardous accidents [1]. With the development of technology, deep learning had been widely used in the field of fault diagnosis for its capability of automatic features extractions and accuracy in diagnosing complex chemical process faults [2].

Bao et al. [3] proposed a fault diagnosis method for chemical processes that combines a deep convolutional neural network (DCNN) with a recurrent neural network (RNN). The application in the Tennessee Eastman (TE) process demonstrated the model's ability to learn dynamic information from raw data in both spatial and temporal domains and achieve high accuracy in identifying abnormal conditions. Lomov et al. [4] proposed to use deep learning architectures, including a novel temporal CNN1D2D model and Generative Adversarial Networks (GANs), for fault detection in the TE process. Souza et al. [5] investigates the development of a fault detection and diagnosis system for TE process using convolutional neural networks (CNNs) and applied transfer learning to address issues of data distribution changes due to alterations in process control strategy. Zhang et al. [6] proposed a fault detection and diagnosis system for complex chemical processes using a bidirectional recurrent neural network (BiRNN) architecture, demonstrating superior performance in fault detection rate (FDR) and fault diagnosis time compared to unidirectional RNNs.

However, the traditional deep learning methods require expert knowledge for hyperparameters tuning to find the optimal hyperparameters for the fault diagnosis

task [7]. This could be time consuming and labour intensive. Liang et al. [8] proposed to use Tree-structured Parzen Estimator (TPE) for hyperparameters optimization in rotating machinery fault diagnosis. Ortiz et al. [9] employed Bayesian optimisation for hyperparameter tuning of machine learning models to achieve high accuracy in fault diagnosis for rotating machinery. Aouichaoui et al. [10] used grid search for hyperparameters tunning for different models and conducted uncertainty estimation methods including bootstrap, ensemble, and dropout to quantify the prediction uncertainty of graph neural networks (GNNs) in chemical property prediction. Nevertheless, research on hyperparameters optimization for fault diagnosis in chemical engineering is limited.

Motivated by the need to overcome the limitations of traditional deep learning-based fault diagnosis approaches which require expert knowledge in tunning hyperparameters, we proposed to utilize MSCNN with Tree-structured Parzen Estimator (TPE) to enhance the performance of fault diagnosis in chemical process. The contributions of this paper are summarized as follows:

1) A novel fault diagnosis framework for chemical process with multi-scale convolutional neural network, leveraging hyperparameters optimization and model ensemble is proposed.

2) Experimental results on TE process indicate the proposed framework outperforms the other comparison methods including PCA, MLP, CNN and MSCNN with default hyperparameters, which highlights the effectiveness of integrating the Multi-Scale module and TPE.

3) Model ensemble technique was conducted to the proposed method, which significantly improved the model performance and provided uncertainty estimations for trustworthy predictions.

The remainder of the paper is organized as follows. Section 2 introduces the methodologies utilized in the proposed method, including the overall proposed workflow, architecture of MSCNN, and introduction of TPE and model ensemble. Section 3 presents the experimental study of the proposed method, using Tennessee Eastman Process (TE process) as case study. Section 4 provides conclusion and future work.

2. METHODOLOGIES

2.1 Proposed workflow

The proposed workflow mainly consists of 4 steps:

1) Data acquisition and data splitting. The fault data are collected from the simulation and divided into train, validation and test sets.

2) Hyperparameters optimization for MSCNN using TPE. The MSCNN model and training and validation sets are initialized with a specific random seed. The TPE runs 500 trial-and-error iterations automatically to find the optimal hyperparameters set that achieves the highest validation accuracy.

3) Models training. The MSCNN model is trained based on the optimal set of hyperparameters that found by TPE. The trained model is saved for further testing.

4) Model performance testing. The trained MSCNN-TPE model predict the fault type of the given test set samples, and the true positive, false positive and false negative are recorded. Based on this, precision, recall and confusion matrix can be presented to evaluate the performance of the model.

2.2 Multi-Scale Convolutional Neural Network

Deep learning has been widely applied in the field of fault diagnosis, with convolutional neural network (CNN) being particularly effective due to their ability to extract hierarchical features. However, traditional CNN often faces challenges in capturing features across multiple scales, which are critical for detecting complex patterns in chemical process data. To address this limitation, a Multi-Scale Convolutional Neural Network (MSCNN) is proposed, as shown in Figure 1.

The key feature of the MSCNN is its ability to extract features at different resolutions through parallel convolutional layers with varying kernel sizes [11]. The one-dimensional fault signal serves as the input to the model, which is processed by a multi-scale convolution module. The module consists of parallel convolutional layer with kernel sizes of 1, 3, and 5. The output of each convolutional layer is computed as follows:

$$z_i^{(j)} = \sum_{k=1}^{K} w_k^{(j)} \cdot x_{i+k-1} + b^{(j)} \quad (1)$$

where $z_i^{(j)}$ is the output of the j^{th} filter at position i, x is the input sequence, $w_k^{(j)}$ represents the weights of the j^{th} filter with kernel size K, and $b^{(j)}$ is the bias term. The weights and bias are trained with gradient decent.

The outputs of multi-scale convolution module are then concatenated, creating a comprehensive feature representation. A maxpooling operation is applied to the concatenated feature maps to reduce dimensionality while retaining important features. The resulting feature maps are further refined by additional convolutional layers. Finally, the features are flattened and passed through fully connected (FC) layers, followed by a softmax output layer to provide the final diagnostic result, represented as predicted probabilities for each fault class.

The MSCNN is trained using cross-entropy loss function:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (2)$$

where C is the number of classes (i.e., fault types). y_i is the true label for class i, taken from the one-hot encoded target vector (1 for the correct class and 0 for others). $\hat{y}_i$ is the predicted probability output by the softmax layer for class i. By minimizing the loss through gradient

decent, the MSCNN achieves optimal fault diagnosis performance.

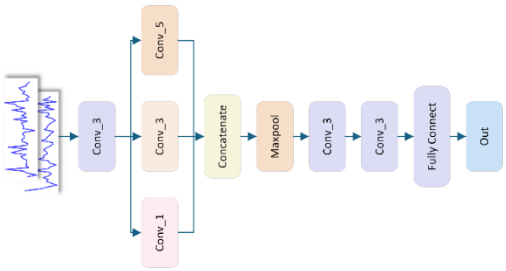

Figure 1: Schematic of the MSCNN structure.

2.3 Hyperparameters Optimization

To train a model that have the best performance on the validation set, we need to iteratively tune the hyperparameters. Instead of tunning them manually, we proposed to use Tree-structured Parzen Estimator, a Bayesian-based hyperparameters optimization method, to search for optimal hyperparameters automatically, as it has been proven effective in [12].

The search space of the hyperparameters and the default values are shown in Table 1. *Lr* refers to the learning rate of the Adam optimizer in the neural network, whose search space follows a *LogUni* type, which means that the value is uniformly distributed in the logarithmic scale. *Dr* is the drop out rate of the FC layer. *Pre_chan* refers to the channel number of the convolutional layer before the multi-scale module. *Post_chan* represents the channels number of the convolutional layer after maxpool. *Hidden_size* stands for the neurons number in FC layer.

Table 1: Hyperparameters Search Space for MSCNN

Name	Value type	Optional value	Default
Lr	LogUni	[0.0001,0.1]	0.01
Dr	LogUni	[0.001,0.5]	0.1
Pre_chan	Choice	8,16,32,64,128,256,512	32
Post_chan	Choice	8,16,32,64,128,256,512	128
Hidden_size	Choice	25,50,100,200,400	100

Tree-structured Parzen Estimator (TPE) [13] is utilized to optimize the hyperparameters in this paper. Historical hyperparameters are grouped to “good” set and “bad” set based on their evaluation value, which is the validation accuracy in this paper. The TPE models the likelihood probability of the two groups as follows:

$$p(x|y) = \begin{cases} l(x), & y > y^{\gamma} \\ g(x), & y \le y^{\gamma} \end{cases} \tag{3}$$

where y is the evaluation value of the hyperparameter x, y^{γ} is the threshold to separate “good” from “bad” observations. By maximizing $l(x)/g(x)$, TPE select the new hyperparameters that have the highest probability to achieve good result relative to a bad one. This strategy allows TPE to search for optimal hyperparameters efficiently compared to manual tunning. In the case study, 500 iterations were conducted, and the searching process took 9 hours and 14 minutes in total.

2.4 Model Ensemble

Instead of training a single model for fault diagnosis, we propose to use model ensemble technique to enhance diagnostic results as well as provide uncertainty estimation. For each model in the ensemble, the training and validation dataset are split differently to introduce diversity in the training data, which is different to the bootstrap sampling strategy [10]. In the meantime, each model is initialized with different parameters initialization, which also creates diversity to the models. With different initialization in both data and model, we enable models to learn from different optimization paths. This approach enhances robustness, as a single model may lack the ability to generalize effectively and is more prone to overfitting the data. By aggregating the predictions of multiple models, the ensemble mitigates these limitations and achieves superior overall performance. After training, the predicted probabilities from individual models are averaged to produce the final output, as shown below:

$$p(\hat{y} = c \mid x) = \frac{1}{M}\sum_{m=1}^{M} p(\hat{y} = c \mid x, \theta_m) \tag{4}$$

where M is the number of models, which is set to 100 in this paper, θ_m is the parameters of the m^{th} model.

3. EXPERIMENTAL STUDIES

3.1 Dataset Description

The Tennessee Eastman (TE) Process is used as a case study to evaluate the performance of the proposed method. This TE Process dataset, originates from a chemical process simulation by the Eastman Chemical Company, is widely used in the field of fault diagnosis and monitoring in chemical process to serve as a benchmark dataset to test process monitoring and fault diagnosis algorithms.

To be specific, the dataset consists of simulation running with 21 different fault types. For each sample, there are 53 process variables in total, but since the 12th variable remains constant, only 52 variables are considered to serve as input data. Due to the length of the paper, the schematic of the TE Process and the details of the faults can be referred to in [14]. There were 26880 samples in total, training, validation and test sets are separated according to the ratio of 6:2:2. For each fault type, there are 768 samples in the training set, 256 samples in the validation set, and 256 samples in the test set.

3.2 Comparison Methods

To highlight the performance of the proposed framework, several comparison methods were carried

Table 2: Results of fault diagnosis on the TE Process. P stands for Precision, R stands for Recall, with values expressed as percentages (%). For each fault type, the highest precision and recall are highlighted in bolded.

	PCA		MLP		CNN		MSCNN		Proposed	
Fault	P	R	P	R	P	R	P	R	P	R
1	100	98.44	87.29	99.22	100	98.83	99.61	**99.22**	**100**	98.83
2	100	96.48	98.03	97.27	100	98.05	99.21	98.44	**100**	**98.83**
3	13.76	17.58	31.40	31.64	43.93	36.72	47.25	**63.67**	**66.39**	61.72
4	82.17	91.80	78.68	98.05	93.23	96.88	**93.80**	94.53	93.31	**98.05**
5	81.82	73.83	94.66	96.88	97.69	99.22	**98.08**	**99.61**	97.31	98.83
6	100	100	100	100	100	100	100	100	**100**	**100**
7	100	100	98.46	100	99.61	99.22	100	100	**100**	**100**
8	93.95	91.02	0	0	92.16	96.48	97.57	94.14	**98.80**	**96.88**
9	19.72	16.41	22.72	35.94	43.44	41.41	57.23	37.11	**61.42**	**60.94**
10	27.94	22.27	61.11	64.45	57.81	86.72	77.57	82.42	**83.15**	**86.72**
11	23.59	26.17	0.00	0.00	72.86	79.69	65.18	79.69	**79.69**	**81.25**
12	86.69	83.98	65.83	92.58	**98.56**	80.47	95.14	91.80	96.39	**93.75**
13	95.65	94.53	78.21	95.31	94.23	95.70	97.23	96.09	**97.65**	**97.27**
14	87.14	82.03	84.21	93.75	**98.40**	96.09	96.51	**97.27**	97.61	95.70
15	17.23	23.83	21.36	24.61	44.63	51.95	51.22	49.22	**57.88**	**66.02**
16	27.72	21.88	46.01	47.27	62.45	55.86	66.52	59.77	**73.33**	**68.75**
17	84.34	92.58	84.00	90.23	91.60	93.75	92.94	92.58	**95.28**	**94.53**
18	96.48	85.55	**99.54**	84.38	95.47	90.63	95.12	91.41	96.71	**91.80**
19	49.55	64.84	66.23	58.98	**78.54**	62.89	76.07	69.53	74.81	**76.56**
20	57.68	60.16	61.09	74.22	82.70	76.56	79.12	76.95	**86.29**	**83.59**
21	56.71	36.33	63.53	63.28	76.98	75.78	74.28	80.08	**78.79**	**81.25**
Avg	66.77	65.70	63.92	68.95	82.11	81.57	83.79	83.50	**87.37**	**87.20**

out for the same diagnosis task. These methods are described as follows:

1) Principal Component Analysis (PCA). A classical dimensionality reduction technique commonly used in machine learning.

2) Multilayer Perceptron (MLP), which is a simple neural network composed of 4 FC layers, with the neurons number of 512, 256, 128, 64 respectively.

3) Convolutional Neural Network (CNN). A deep learning model adapted by replacing the multiscale module in MSCNN with two sequential convolutional layers, each with a kernel size of 3.

4) MSCNN. The MSCNN model trained with the default hyperparameters shown in Table 1. This model utilizes multiscale features to improve diagnostic performance.

5) MSCNN-TPE (Proposed). The MSCNN model trained with the optimal hyperparameters searched by TPE, which is the proposed method.

The experiments were conducted on a cluster server equipped with an AMD EPYC 9354 CPU and an Nvidia Volta 100 GPU. The models are implemented with the python packages scikit-learn and Pytorch 2.5. The optimizer used was Adam, with a learning rate set to 0.01. A learning rate scheduler was applied, reducing the learning rate by 10% every 10 epochs. Early stopping was employed with a patience of 10, meaning the training process would stop if the validation loss did not decrease for 10 consecutive epochs. By using consistent settings across all methods, this setup ensures a fair comparison between the approaches.

3.3 Results and Discussion

3.3.1 Evaluation Metrics

To evaluate the performance of the fault diagnosis models, we use precision (P) and recall (R) as the key metrics. These are particularly important in scenarios like fault diagnosis, where both false positives and false negatives can have critical consequences. The calculation of precision and recall are calculated as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{4}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{5}$$

True Positive (TP) refers to a positive sample being correctly classified. False Positive (FP) represents a negative sample but misclassified as positive. False Negative (FN) indicates a positive sample that is misclassified as negative. In plain words, precision measures the proportion of correctly identified positive cases among all predicted positive cases, while recall measures the proportion of actual positive cases that the model successfully identified.

3.3.2 Results and Discussion

Table 2 shows the overall diagnostic results of the experiment. Some conclusions can be drawn from it:

Limited competitiveness of PCA, MLP and CNN. Both PCA, MLP and CNN have poor precision and recall in many of the fault types, especially on fault type 3,9,11 and 15. Among them, PCA performs the worst, followed by MLP, while CNN shows slightly better results. It is also worth noting that MLP fails to detect fault type 8 and 11 entirely, likely due to its architecture limitations in learning distinguishable features in complex high-dimensional feature spaces.

Effectiveness of the Multi-Scale module in MSCNN. Compared to CNN, MSCNN demonstrates

improvements in both precision and recall, particularly for some specific fault types. The average precision and recall of MSCNN (83.79%, and 83.50% respectively) are higher than those of CNN (82.11% and 81.57%), indicating the effectiveness of the integration of multi-scale module.

Effectiveness of TPE. The proposed method outperformed all comparison methods in terms of both precision and recall in most of the fault types, indicating its superiority. Specifically, the proposed method achieves an average precision and recall improvement of 3.58% and 3.70% respectively, over MSCNN, indicating the effectiveness of TPE in enhancing model performance.

Comparison of training loss. The training and validation losses of different approaches are shown as Figure 2. MLP exhibits the highest training and validation losses, and they can't decrease effectively after the longest training epoch, indicating the insufficient capacity of MLP to extract meaningful features from the data. CNN and MSCNN achieve similar losses, while the validation loss of MSCNN is slightly lower. On the other hand, the proposed method achieves the lowest losses, demonstrating its superior performance on both the training and validation sets, which aligns with the results observed on the test set.

Analyzing of confusion matrix. Due to the space limitations, only the confusion matrix of the proposed method is shown in Figure 3. By looking into the confusion matrix, we can have an intuitive analysis about the performance of the model and have insights into which fault types are more challenging to classify and how they are confused with one another.

Overall, the proposed method exhibits a higher number of correctly classified samples along the diagonal compared to MSCNN. Specifically, The TP of fault types 9 and 15 are 156 and 169 respectively— significantly higher than those of MSCNN (95 and 126 respectively). This improvement highlights the proposed method's ability to accurately classify faults. This result also indicated that some specific fault types are more difficult to classify and could be easily confused with one another. The reason could be their similar fault patterns, which make their feature representations highly alike. For example, fault 3 (D Feed Temperature (Stream 2) Step) and fault 9 (D Feed Temperature (Stream 2) Random Variation) exhibits similar characteristics, leading to increased misclassification between them.

Significance of introduction of model ensemble. As shown in Table 3 are diagnostic results of some difficult faults of the proposed method before and after using model ensemble. The model number is 100 in the ensemble. We can observe a significant improvement in both precision and recall after model ensemble, highlighting its effectiveness. Moreover, the uncertainty estimation of a fault sample can be calculated from the entropy of the probability distribution, and additional analyses can be provided by looking at the standard deviations, mean and other statistics from the 100 models in the ensemble. This approach provides more reliable result compared to using the prediction from a single model, as it captures the diversity in both model initializations and data splits.

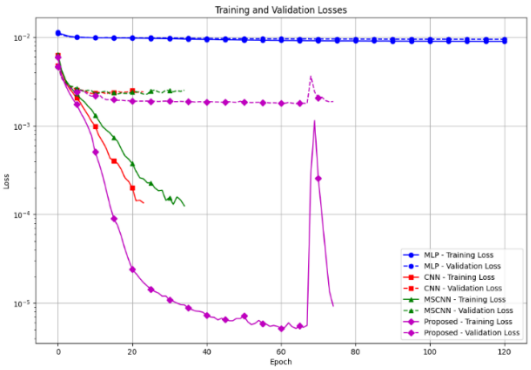

Figure 2: Training and validation losses of different approaches.

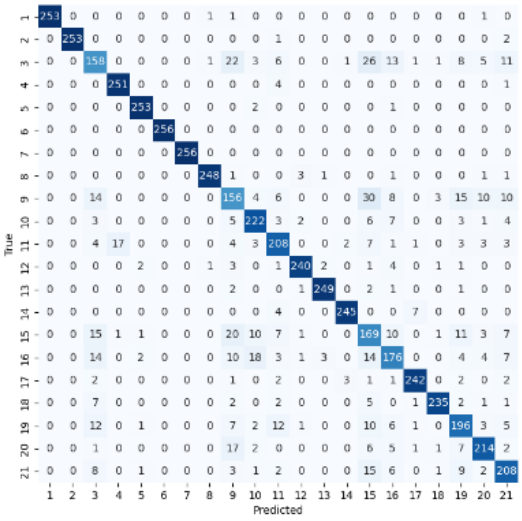

Figure 3: Confusion matrix of the proposed method.

Table 3: Results before and after model ensemble for difficult faults

Fault	Proposed-Single		Proposed-Ensembled	
	P	R	P	R
3	66.39	61.72	66.79	69.92
9	61.42	60.94	71.85	75.78
11	79.69	81.25	85.43	82.42
15	57.88	66.02	66.08	73.88
19	74.81	76.56	78.20	81.25
Avg	68.04	69.30	73.67	76.65

4. CONCLUSION AND FUTURE WORK

To conclude, this paper proposed a fault diagnosis framework that integrates a multi-scale convolutional neural network with hyperparameters optimization and model ensemble to enhance fault diagnosis for complex chemical processes.

Experimental results on TE process indicate the proposed framework outperforms the other comparison methods including PCA, MLP, CNN and MSCNN with default hyperparameters. This shows the effectiveness of the introduction of Multi-Scale module and TPE. Additionally, model ensemble was applied to further improve the fault diagnosis performance and provide uncertainty estimations, enabling more reliable decision-making.

For future study, we aim to further explore model ensemble techniques, compare different models and uncertainty estimation methods, and contribute to trustworthy fault diagnosis that not only focus on the accuracy but also the reliability. Model pruning [15] will also be investigated to reduce model complexity while maintaining a comparable diagnostic performance.

ACKNOWLEDGEMENTS

This research was funded by the European Union Horizon Europe 2022 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No. 101119358 (ProSafe).

REFERENCES

1. Bi X, Qin R, Wu D, Zheng S, Zhao J. One step forward for smart chemical process fault detection and diagnosis. Comput Chem Eng 2022;164:107884. https://doi.org/10.1016/j.compchemeng.2022.107884.
2. Md Nor N, Che Hassan CR, Hussain MA. A review of data-driven fault detection and diagnosis methods: applications in chemical process systems. Rev Chem Eng 2020;36:513–53. https://doi.org/10.1515/revce-2017-0069.
3. Bao Y, Wang B, Guo P, Wang J. Chemical process fault diagnosis based on a combined deep learning method. Can J Chem Eng 2022;100:54–66. https://doi.org/10.1002/cjce.24153.
4. Lomov I, Lyubimov M, Makarov I, Zhukov LE. Fault detection in Tennessee Eastman process with temporal deep learning models. J Ind Inf Integr 2021;23:100216. https://doi.org/10.1016/j.jii.2021.100216.
5. e Souza ACO, de Souza MB, da Silva FV. Enhancing fault detection and diagnosis systems for a chemical process: a study on convolutional neural networks and transfer learning. Evol Syst 2024;15:611–33. https://doi.org/10.1007/s12530-023-09523-y.
6. Zhang S, Bi K, Qiu T. Bidirectional Recurrent Neural Network-Based Chemical Process Fault Diagnosis. Ind Eng Chem Res 2020;59:824–34. https://doi.org/10.1021/acs.iecr.9b05885.
7. Qin R, Zhao J. Adaptive multiscale convolutional neural network model for chemical process fault diagnosis. Chin J Chem Eng 2022;50:398–411. https://doi.org/10.1016/j.cjche.2022.10.001.
8. Liang J, Liao Y, Chen Z, Lin H, Jin G, Gryllias K, et al. Intelligent fault diagnosis of rotating machinery using lightweight network with modified tree-structured parzen estimators. IET Collab Intell Manuf 2022;4:194–207. https://doi.org/10.1049/cim2.12055.
9. Ortiz R, Miranda-Chiquito P, Encalada-Davila A, Marquez LE, Tutiven C, Chatzi E, et al. An Enhanced Modeling Framework for Bearing Fault Simulation and Machine Learning-Based Identification With Bayesian-Optimized Hyperparameter Tuning. J Comput Inf Sci Eng 2024;24:091002. https://doi.org/10.1115/1.4065777.
10. Aouichaoui ARN, Mansouri SS, Abildskov J, Sin G. Uncertainty estimation in deep learning-based property models: Graph neural networks applied to the critical properties. AIChE J 2022;68:e17696. https://doi.org/10.1002/aic.17696.
11. Song Q, Jiang P. A multi-scale convolutional neural network-based fault diagnosis model for complex chemical processes. Process Saf Environ Prot 2022;159:575–84. https://doi.org/10.1016/j.psep.2021.11.020.
12. Loshchilov I, Hutter F. CMA-ES for hyperparameter optimization of deep neural networks. arXiv preprint 2016; arXiv:1604.07269. https://doi.org/10.48550/arXiv.1604.07269.
13. Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for Hyper-Parameter Optimization. Adv. Neural Inf. Process. Syst., vol. 24, Curran Associates, Inc.; 2011.
14. Melo A, Câmara MM, Clavijo N, Pinto JC. Open benchmarks for assessment of process monitoring and fault diagnosis techniques: A review and critical analysis. Comput Chem Eng 2022;165:107964. https://doi.org/10.1016/j.compchemeng.2022.107964.
15. Ruan D, et al. Light convolutional neural network by neural architecture search and model pruning for bearing fault diagnosis and remaining useful life prediction. Sci Rep 2023;13:5484. https://doi.org/10.1038/s41598-023-31532-9.