Industrial Big Data Analytical System in Industrial Cyber-Physical Systems Based on Coarse-to-Fine Deep Network

Ruonan Liu, *Member, IEEE*, Quanhu Zhang, Yu Wang, *Member, IEEE*, Zengxiang Li, Dongyue Chen, Steven X. Ding, Qinghua Hu, *Senior Member, IEEE*, and Boyuan Yang, *Member, IEEE*

***Abstract*—In smart factories, there have been increasing requirements for industrial Big Data analysis of complex systems. With the rapid development of industrial cyber-physical systems (ICPS) and communication techniques, the scale and complexity of industrial data are growing explosively, which not only provides massive operational information of industrial systems but also brings challenges in Big Data analysis. In this paper, to overcome the intra/inter-class distance unbalance and local minima problems in traditional deep learning-based methods, an industrial Big Data analytical system based on a coarse-to-fine network (CTFN) is proposed for intelligent industrial Big Data analysis and condition monitoring of complex system. In addition, considering the gap between semantic comprehension and natural characteristics of different failures, a structure learning algorithm is proposed to get rid of the complicated hyper-parameters and implement intelligentization authentically. Finally, an experimental verification was carried on a nuclear power system dataset with 362,994 samples from 66 fault categories. The results demonstrate the effectiveness and superiority of the proposed method in condition monitoring of industrial systems, which provides a promising tool for industrial Big Data analysis in ICPS.**

***Index Terms*—Coarse-to-fine network, industrial Big Data, industrial cyber-physical systems, large-scale condition monitoring of complex systems, structure learning.**

Manuscript received 16 January 2023; revised 28 April 2023; accepted 27 October 2023. Date of publication 9 November 2023; date of current version 22 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62106174 and 62206199, in part by the Alexander von Humboldt Foundation, and in part by the Open Research Fund of State Key Laboratory of High Performance Complex Manufacturing, Central South University under Grant Kfkt2022-10. *(Corresponding author: Yu Wang.)*

Ruonan Liu, Quanhu Zhang, Yu Wang, Dongyue Chen, and Qinghua Hu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: ruonan.liu@tju.edu.cn; quanhuzhang@tju.edu.cn; armstrong_wangyu@tju.edu.cn; dyue_chen@163.com; huqinghua@tju.edu.cn).

Zengxiang Li is with the Digital Research Institute, ENN Group, Langfang, Hebei 065001, China (e-mail: lizengxiang@enn.cn).

Steven X. Ding is with the School of Automation, University of Duisburg-Essen, 47057 Duisburg, Germany (e-mail: steven.ding@uni-due.de).

Boyuan Yang is with the College of Artificial Intelligence, Nankai University, Tianjin 300071, China (e-mail: yby@nankai.edu.cn).

Digital Object Identifier 10.1109/TICPS.2023.3331331

I. Introduction

With the rapid development of manufacturing, industrial systems not only can produce more business value but also are becoming increasingly complex. An untimely response to a fault may trigger a cascading failure and large-scale blackout, even catastrophic accidents [1]. As an effective tool to ensure production efficiency and operation safety, operational data analysis and condition monitoring are very important to the industrial cyber-physical systems (ICPS) [2]. Traditional operational data analysis methods recognize different fault types based on the extracted features via various signal processing techniques. In recent years, with the development of sensing techniques, a large amount of industrial data has been produced from monitoring systems [3], which makes data-driven condition monitoring possible. Among them, deep learning (DL) methods aim at automatically learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower-level features, thereby showing the potential to make the industrial system more intelligent [4]. Several branch DL methods have already demonstrated state-of-the-art performance in operational data analysis tasks, including autoencoder [5], recurrent neural network (RNN) [6], deep belief network (DBN) [7] and the most widely used convolutional neural network (CNN) [8]. Considering that learning algorithms can be a powerful tool for improving ICPS network performance, J. Wang et al. summarized the development of learning-aided wireless networks by investigating a series of popular learning algorithms in ICPS, providing some specific examples and a range of promising open issues in the future [9].

As effective as DL methods are, they still suffer from a critical problem. In literature, state-of-the-art DL-based diagnosis methods performed well on the big dataset with several fault types of a single or a few components, including bearings [10], motors [11], gearboxes [12], and so on. However, in Industry 4.0, the mechanical equipment in the modern industry has been increasingly functional and complicated [13]. Take SIEMENS SGT-800 gas turbine as an example. It consists of an inlet, compressor, combustor, turbine, exhaust, and other components, which can lead to an enormous variety of faults. Therefore, the era of Big Data in industrial systems does not only mean a mass of data but also represents a large number of fault

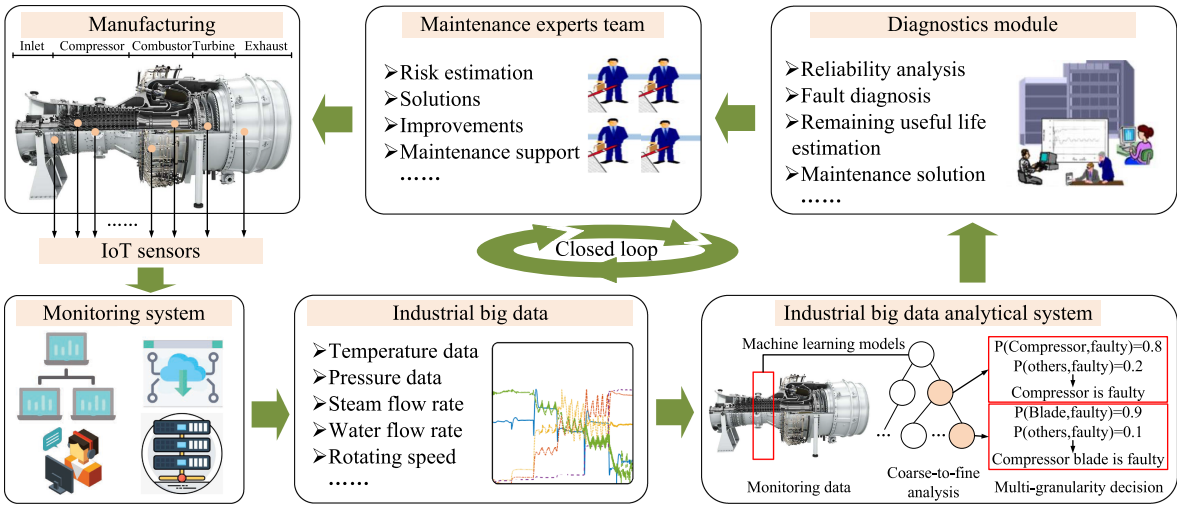


Fig. 1. Coarse-to-fine network for industrial Big Data analysis in ICPS.

types. Although DL-based methods can be a promising tool for analyzing a large number of data with several fault types, their performance will decrease rapidly when facing dozens or hundreds of fault types. The reasons are as follows.

In traditional DL methods, different faults are shared with one deep network for feature extraction and fault recognition. This design can be effective when diagnosing the faults of a single component but may show drawbacks when dealing with a lot of components in operational data analysis of complex systems. Because of the increasing diversity of different failures will lead to the intra/inter-class distance unbalance problem in feature space [14]. That is, the feature distance between similar failures of one component is small and hard to distinguish, while the feature distance between failures in different components is significantly different and can be easily classified. As a result, the feature inter-class distance of some similar failures can be even smaller than the intra-class distance of some failures, which may push the learning process away from the global optimum. Meanwhile, when faced with large-scale operational data analysis tasks, a lot of faults will lead to the exponential growth of parameters and structure complexity, which can cause a serious local minima problem in deep networks [15]. Therefore, the large-scale operational data analysis of complex systems with various fault types not only is a typical class of mission-critical industrial automation applications and in urgent need for the development of ICPS and smart factory [16], but also is a hard nut to crack due to the intra/inter-class distance unbalance and local minima problems.

To overcome the above problems, this paper proposes an intelligent industrial Big Data analytical system based on a CTFN model, which is able to perform industrial Big Data analysis of industrial systems intelligently in a divide-and-conquer strategy, thus shows great potential in ICPS and smart factories development, as shown in Fig. 1. The main contributions of this article are summarized as follows.

1) An industrial Big Data analytical system based on a CTFN is proposed. Instead of diagnosing the faults of one or few components, the proposed method aims to perform the industrial Big Data analysis of a complete industrial system intelligently. By spreading faults into different coarse nodes, the intra/inter-class distance unbalance problem can be avoided. And due to the decrease of the fault number in one coarse task, the local minima problem can also be mitigated. Thus, the CTFN architecture makes it possible to address the large-scale diagnosis task intelligently.
2) The design of the coarse-to-fine structure for such a large-scale diagnosis task not only is energy-consuming and time-wasted, but also needs enough prior knowledge. Therefore, a data-driven structure learning algorithm is proposed to extract the coarse-to-fine knowledge and design an appropriate CTFN structure automatically and intelligently.
3) A large-scale condition dataset of a nuclear power system is collected for experimental verification of the proposed method, which contains 362,994 samples from 66 fault categories. The experimental results and comparison with six state-of-the-art diagnosis methods demonstrated that the proposed method can be a promising tool for a large-scale diagnosis task in an industrial Big Data environment.

The remaining parts of this paper are organized as follows. The proposed method is elaborated in Section II. In Section III, the effectiveness and superiority of the proposed method are validated on an operational data analysis scenario of a nuclear power system. Finally, Section IV concludes this paper.

II. Problem Formulation

The existing deep learning-based methods usually directly calculate the finest granularity diagnostic result by modeling a fault classifier from a global view, while ignoring the physical property of each failure. For instance, given a monitoring dataset of a nuclear power system, it is relatively easy to tell if there

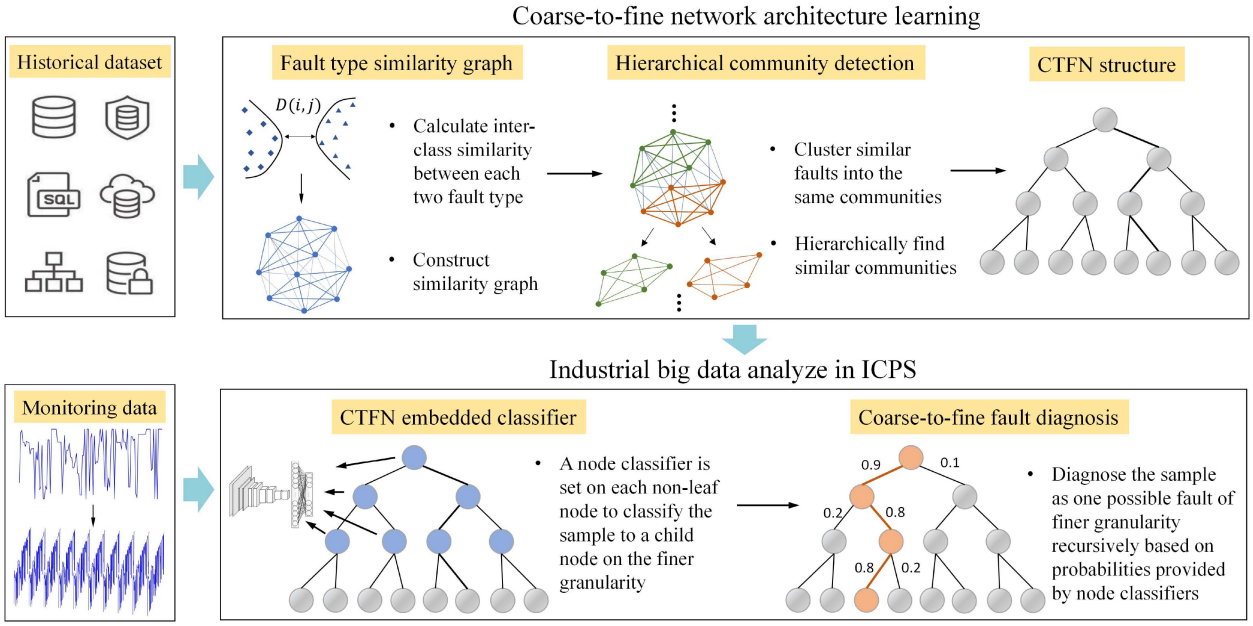


Fig. 2. Framework of the proposed CTFN for industrial Big Data analysis.

is a fault in the secondary loop according to the indexes or distinguish if it is a heater fault or a pump fault in the secondary loop, but it can be much harder to diagnose a finer failure directly, such as diagnosing if there is a gas inlet valve failure or a regulating valve failure.

Such a difficult problem can be divided into several simple sub-problems in different granularity, and the final diagnosis result can be obtained by performing a coarse-to-fine way along with a CTFN architecture of fault types. In this way, a sample is first diagnosed as a coarse-grained fault, e.g., the secondary loop failure, and then identified as a fault of finer granularity, e.g., gas inlet valve failure, until an expected fault type is found.

To achieve this goal, the proposed approach first learns a coarse-to-fine architecture that consists of all fault types and their coarse-granularity fault concepts; subsequently, different node classifiers are trained and embedded in each non-leaf node of the structure to identify the possible faults in their granularity, which makes up a CTFN; finally, the CTFN makes predictions for the test sample along with the structure in a coarse-to-fine way, where the sample starts from the root node and diagnosed as a child node of the current node recursively until it is diagnosed as a leaf-node fault.

III. The Proposed Approach

The objective of this article is to explore and exploit the multi-granularity relations of various fault types to address the industrial Big Data analysis problem for industrial complex systems. Formally, given training samples $\{\{\boldsymbol{x}_i^j\}_{i=1}^{M_i}\}_{j=1}^{N}$ in N fault types, where $\boldsymbol{x}_i^j$ is the j-th sample in the i-th fault type; M_i is the number of samples in the i-th category; $i \in \{1, 2, \ldots, M_i\}$, and $j \in \{1, 2, \ldots, N\}$, the objective of the proposed framework is to learn a CTFN architecture $\boldsymbol{T}$, to train the CTFN classifier $\boldsymbol{W}$, and to make predictions along $\boldsymbol{T}$ in a coarse-to-fine way to obtain leaf-node diagnosis results of test samples $\{\hat{\boldsymbol{y}}_l\}_{l=1}^{t}$, where t is the number of test samples.

The framework of the proposed CTFN approach has two components: CTFN architecture learning and CTFN embedded diagnosis (See Fig. 2). In the former, an affinity matrix is first constructed to measure the inter-class relations between different fault categories. Subsequently, parameter-free hierarchical community detection is designed to efficiently learn the CTFN architecture of fault categories. In the latter, the learned structure is embedded into the classification process by learning multiple classifiers on nodes of the structure. During the diagnosis procedure, samples are identified from the coarse-grained nodes to the very fine-grained nodes.

A. Coarse-to-Fine Network

CTFN organizes classes into a tree-like graph from the coarse-grained to the fine-grained. There are two kinds of structures in class hierarchy, tree and directed acyclic graph (DAG). The tree structure is utilized in this work because it is the most common and widely used (the CTFN architecture represents a tree structure in this paper).

A tree hierarchy of the class labels represents a kind of "IS-A" relationship between labels [17]. Specifically, Kosmopoulos et al. pointed out that the properties of "IS-A" relationship can be described as asymmetry, anti-reflexivity, and transitivity [18]. Define a tree as a triplet $(\mathcal{V}, \mathcal{E}, \prec)$ with a set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, where $\mathcal{E}$ represents a set of edges between nodes in different levels, and $\prec$ represents the parent-children relationship between nodes connected by edges ("IS-A" relationship), formulated as follows:

1) Asymmetry: if $v_i \prec v_j$ then $v_j \not\prec v_i$ for $\forall v_i, v_j \in \mathcal{V}$;

2) Anti-reflexivity: $v_i \prec v_i$ for $\forall v_i \in \mathcal{V}$;
3) Transitivity: if $v_i \prec v_j$ and $v_j \prec v_k$, then $v_i \prec v_k$ for $\forall v_i, v_j, v_k \in \mathcal{V}$.

These properties result in the fact that an arbitrary node has only one parent node except for the root node, which is unique and on the top of a certain tree CTFN structure. Besides the root node, nodes in the tree CTFN structure can be generally categorized into several types. Typically,

1) A leaf node is at the bottom of a tree CTFN structure and has no child nodes;
2) An internal node is an arbitrary node that is not the root node or a leaf node;
3) A sibling node is the node that has the same parent;
4) An ancestor node is the node that has more than one-level parent relationship with the current node;
5) The decedent nodes of a certain node are its corresponding leaf nodes.

B. CTFN Architecture Learning

In order to deal with the complex failure relationship with different separability in the Big Data environment, the large-scale diagnostic task is divided into several sub-diagnostic tasks. If this coarse-to-fine and dendritical structure is only constructed based on human knowledge, the semantic gap problem will be a great barrier to improving performance. That is, when constructing the CTFN architecture manually, the similarity of different faults is measured by the semantic comprehension similarity of different failures, instead of the natural characteristics of the industrial data. As a result, the constructed CTFN architecture can be easier for understanding, but harder for operational data analysis and condition monitoring.

CTFN organizes classes into a tree-like graph from the coarse-grained to the fine-grained. To automatically obtain CTFN, a data-driven architecture learning method is also proposed in this paper to address the above-mentioned problems.

1) Affinity Matrix Construction: The affinity matrix measures the relation between every two categories and represents all the relations as a weighted graph. The key point of affinity matrix construction is to properly calculate the distance or similarity of each two categories. Intuitively, the inter-class similarity can be measured by calculating the similarity of all the sample pairs of both categories, but this is computationally unacceptable for large-scale tasks. Recently, Fan et al. used the mean vector of samples to represent all samples of a category and computed the inter-class similarity by the distance between the mean vectors [19]. Although this method is efficient, it is hard to describe the information of a category by only one sample vector. Inspired by [20], an effective and efficient inter-class measurement is applied to take high-order information of samples into consideration.

Suppose $\{\boldsymbol{x}_l^i\}_{l=1}^{N_i}$ are N_i samples in the i-th fault category C_i, where $C = \{C_1, C_2, \ldots, C_N\}$ is the set of all the fault categories. The distance between each two categories can be formulated as

$$\mathcal{D}(C_i, C_j) = \sqrt{\frac{1}{N_i N_j} \sum_s \sum_t \left\| \boldsymbol{x}_s^i - \boldsymbol{x}_t^j \right\|^2}, \tag{1}$$

where $s \in \{1, \ldots, N_i\}$ and $t \in \{1, \ldots, N_j\}$. (1) requires $N_i \times N_j$ norm operations, which has a high computational cost. This equation can be transformed to reduce the computation as

$$\mathcal{D}^2(C_i, C_j) = \frac{1}{N_i N_j} \sum_s \sum_t \left\| (Q_i - \Delta \boldsymbol{x}_s^i) - (Q_j - \Delta \boldsymbol{x}_t^j) \right\|^2, \tag{2}$$

where $Q_i = \frac{1}{N_i} \sum_{l=1}^{N_i} \boldsymbol{x}_l^i$ is the mean vector of the i-th category and $\Delta \boldsymbol{x}_s^i = Q_i - \boldsymbol{x}_s^i$. Note that there is a property of the mean vector: $\sum_{s=1}^{N_i} \boldsymbol{x}_s^i = 0$. Based on this property, the inter-class distance measurement can be obtained as follows.

$$\begin{aligned}
\mathcal{D}^2(C_i, C_j) &= \frac{1}{N_i N_j} \sum_s \sum_t \left\| (Q_i - \Delta \boldsymbol{x}_s^i) - \left(Q_j - \Delta \boldsymbol{x}_t^j\right) \right\|^2 \\
&= \|Q_i - Q_j\| - \frac{2\|Q_i - Q_j\|}{N_i N_j} \sum_{s=1}^{N_i} \sum_{s=1}^{N_j} \left(\Delta \boldsymbol{x}_s^i \Delta \boldsymbol{x}_t^j\right) \\
&\quad + \frac{1}{N_i N_j} \sum_{s=1}^{N_i} \left(\Delta \boldsymbol{x}_s^i \Delta \boldsymbol{x}_t^j\right)^2 \\
&= \|Q_i - Q_j\|^2 + \frac{1}{N_i} \left(\Delta \boldsymbol{x}_s^i\right)^2 + \frac{1}{N_j} \left(\Delta \boldsymbol{x}_t^j\right)^2 \\
&= \|Q_i - Q_j\|^2 + \sigma_i^2 + \sigma_j^2,
\end{aligned} \tag{3}$$

where $\sigma_i^2 = \sum_{s=1}^{N_i} \frac{1}{N_i} (Q_i - \boldsymbol{x}_s^i)^2$ is the variance of the i-th category. (3) reduces the norm operations to only $(N_i + N_j + 1)$ in comparison with $(N_i \times N_j)$ classical distance computation in (1). Additionally, it contains high-order statistical information and thus can better describe the information of every category.

To build the affinity matrix, the distance-based measurement should be transformed into a similarity-based measurement. The commonly used Gaussian transformation is leveraged based on results of (3), and thus the element of the affinity matrix can be obtained as

$$\boldsymbol{A}_{ij} = \exp\left(-\frac{\mathcal{D}(C_i, C_j)}{\delta_{ij}}\right), \tag{4}$$

where σ_{ij} is the scaling factor.

2) Hierarchical Community Detection: With the affinity matrix, the CTFN architecture can be obtained by grouping similar categories recursively. In artificial intelligence, researchers apply hierarchical spectral clustering to build the CTFN architecture [19], [20]. The construction of the CTFN architecture can be divided into a top-down Bayesian process that clusters the categories based on the previous grouping results recursively. Specifically, given the affinity matrix $\boldsymbol{A}$ in (4), spectral clustering is first performed to group all the possible N categories into θ groups $= \{C_1', C_2', \ldots, C_i', \ldots, C_\theta'\}$. Then, for all the new groups $\{C_i'\}_{i=1}^{\theta}$, the spectral clustering algorithm [21] is applied to all the decedent nodes of C_i' again to generate another θ groups, respectively. The process is terminated if it meets one of the following conditions: (1) The number of the decedent nodes of C_i' is no more than θ; (2) The tree depth is larger than the pre-defined maximum tree depth.

Although this method can build the CTFN architecture flexibly, it requires two manually tuned parameters, the number of child nodes per parent θ and tree depth Φ, which is difficult

and time-consuming to set [22]. Suppose a given industrial Big Data analysis task has N fault categories, the minimum value of $\theta_{\min}$ is easily obtained as 2. The maximum value of $\theta_{\max}$ should lead to the structure that has the largest number of superordinate groups while each superordinate group has the minimum number of child nodes. To this end, $\theta_{\max}$ should be $N/2$ because the number of child nodes is 2 for each superordinate group, so the possible range for θ is $[2, N/2]$. Similarly, another parameter tree depth Φ ranges from $[2, (2N-1)/2]$. To fully explore the combinations of the parameters, users have to compute $O(N^2)$ attempts to obtain the final results, and this is unacceptable if the category number N is very large even if the search space can be eliminated empirically. In this regard, a parameter-free method is used to optimize the modularity of the community. Here, the fast community detection algorithm [23] is improved for hierarchical community detection. For the flat version of community detection, the key point is to compute the quality or the modularity Q_h of the h-th level. However, in the coarse-to-fine diagnosis task scenario, Q_h is needed to be computed in the structure:

$$Q_h = \frac{1}{2b}\sum_{i,j}\left(\boldsymbol{A}_h(i,j) - \frac{k_i k_j}{2b}\right)\delta(c_i, c_j), \tag{5}$$

where $\boldsymbol{A}_h$ is the inter-class similarity matrix of the h-th level, $\boldsymbol{A}_h(i,j)$ is the similarity degree of the edge between class i and class j, $k_i = \sum_j \boldsymbol{A}_{ij}$ is the sum of similarity degrees of the edges attached to the vertex class i, c_i is the community to which vertex class i is assigned, and the $\delta(p,q)$ function is 1 if $p = q$ and 0 otherwise, and $b = \frac{1}{2}\sum_{ij}\boldsymbol{A}_{ij}$. Then, following [23], the local modularity changes and the aggregated community are optimized at each node v. Specifically, the gain ΔQ_h of the modularity can be evaluated by removing the i-th class and placing it in the j-th community, which is the neighbor of i. If this gain is positive, node i is then placed in the community for which this gain is maximum and stays in its original community otherwise. This process is applied repeatedly and sequentially until no further improvement can be achieved. By applying the algorithm recursively, the CTFN architecture can be obtained.

$$\begin{aligned}\Delta Q_h = &\left[\frac{\sum_{in} S_h(\boldsymbol{A}_h) + 2k_{i,in}}{2b} - \left(\frac{\sum_{tot} S_h(\boldsymbol{A}_h) + k_i}{2b}\right)^2\right] \\ &- \left[\frac{\sum_{in} S_h(\boldsymbol{A}_h)}{2b} - \left(\frac{\sum_{tot} S_h(\boldsymbol{A}_h)}{2b}\right)^2 - \left(\frac{k_i}{2b}\right)^2\right],\end{aligned} \tag{6}$$

where S_h is the community in the h-th level, $\boldsymbol{A}_h$ is the similarity matrix computed by (3) and (4), $\sum_{in} S_h(\boldsymbol{A}_h)$ is the sum of the weights of the links inside S_h of which relations between nodes are constructed by $\boldsymbol{A}_h$, $\sum_{tot} S_h$ is the sum of the weights of the links incident to nodes in S_h, k_i is the sum of weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in S_h, and b is the sum of the weights of all the links in the network. In this way, the optimal CTFN can be automatically obtained without any prior knowledge.

C. CTFN Embedded Industrial Big Data Analysis

After obtaining the learned CTFN architecture, it should be embedded into the classification process for effective and efficient industrial Big Data analysis. Specifically, a classifier on each non-leaf node of the tree-shaped CTFN should be trained during training; in the diagnosis process, a top-down manner is employed by recursively diagnosing the collected data to the child nodes with the largest probability, starting from the root node to the leaf nodes [24]. Each classifier is responsible for a certain subset of categories, so the original difficult problem is decomposed into several simple sub-problems on different nodes and thus can result in more accurate diagnosis results.

Suppose $\hat{\boldsymbol{x}}_l$ is the l-th test sample, where $l \in \{1, \ldots, t\}$, $\boldsymbol{W}$ is the set of trained node classifiers, $\boldsymbol{w}_v \in \boldsymbol{W}$ is the trained classifier at node $v \in V$, the probability of sample $\hat{\boldsymbol{x}}_l$ belonging to i-th child node of v is defined as

$$P_v^i(\hat{\boldsymbol{x}}_l) = (\hat{\boldsymbol{x}}_l)^T \boldsymbol{w}_v \boldsymbol{I}(i), \tag{7}$$

where $\boldsymbol{I}(i)$ is a one-hot vector whose i-th element is 1 and others are 0. To decide which child node of node v to be assigned for the sample, a greedy algorithm that selects the child node with the largest probability is used at each node:

$$d_v = \arg\max_i \left(P_v^i(\hat{\boldsymbol{x}}_l)\right). \tag{8}$$

The test sample $\hat{\boldsymbol{x}}_l$ is recursively classified to the child node with the largest probability d_v until a leaf node is reached. In this way, more accurate diagnosis results can be achieved by solving a series of simple sub-problems instead of the original difficult problem. Such a top-down and divide-and-conquer classification can lead to higher efficiency for conventional base classifiers, such as logistic regression and SVM. Moreover, the proposed method can reduce the computational complexity from $O(N)$ to $O(\log N)$. Suppose that the base classifier uses the logistic regression, whose transformed version softmax is common in DNNs. For a conventional flat classifier, a multi-class classification problem is transformed into multiple binary classification sub-problems by constructing N one-versus-rest sub-classifiers. In terms of fault types N, the diagnosis complexity of a conventional flat classifier is $O(N)$. In contrast, the coarse-to-fine classifier divides N fault types into several sub-groups, wherein each sub-group contains fewer fault types. On each granularity, only a part of fault types is considered, and this results in a diagnosis complexity of $O(\log N)$ [25]. Therefore, the proposed approach can not only benefit the diagnosis performance but also the diagnosis efficiency. However, it is worth noting that the testing time of the proposed approach will be longer than the corresponding flat approach if a deep learning-based model is used on each node. The reason may mainly be that the computations on layers before the softmax layer are duplicate for all base models, so this will lead to more diagnosis time for their coarse-to-fine versions.

It is worth mentioning that refined data analysis and precise fault location can be done with the proposed CTFN method. On one hand, data are analyzed in different granularity of the learned structure, thus providing general analysis on high levels

and refined analysis on low levels. On the other hand, since fault data are analyzed from the coarse to the fine, multi-granularity location results can be obtained with the proposed system. For example, supposing a heater fault occurs in the secondary loop component, and it is first recognized as faults in the secondary loop and then identified as a heater fault. Such a multi-level fault location can be obtained, which provides users with more flexible results for various scenarios.

D. Unknown Faults Detection

Conventionally, the methods of industrial Big Data analysis are assumed to know all the fault types in the training phase. In real-world scenarios, this assumption does not always hold. Therefore, the system is supposed to detect unknown faults that do not occur during training. A straightforward way for this issue is to set a certain threshold τ, which is dependent on the requirement of the target task, and to regard the sample as an unknown if its maximal prediction score is lower than τ. For the CTFN structure, this corresponds to the decision on the leaf node level.

$$\Omega_v(\boldsymbol{x}) = \begin{cases} 0 & \max\left(P_v^i(\hat{\boldsymbol{x}}_l)\right) < \tau \\ d_v & \max\left(P_v^i(\hat{\boldsymbol{x}}_l)\right) \geq \tau. \end{cases} \tag{9}$$

Considering the top-down prediction process, a local decision can be obtained on each granularity. In the perspective of detecting the unknown, the CTFN structure can judge whether the sample belongs to the known fault types or the unknown type on different granularity. In this scenario, decisions on coarse granularity can be seen as auxiliary information for leaf-node decisions, and properly combining such multi-granularity decisions can be helpful for detecting unknowns. Specifically, the final decision score d^A is adjusted by attaching an importance factor to each decision of a certain granularity: $d^A = \sum_{i+1}^{\Phi} \alpha_i d^i$, where Φ is the depth of the structure, $\sum_i \alpha_i = 1$, and α_i is the importance factor of the decision on the i-th granularity.

IV. Experimental Study

The ICPS is a network of intelligent and highly connected industrial components, including a variety of core industrial software, industrial Internet of Things platform and terminal equipment, edge computing equipment, industrial artificial intelligence applications, the underlying real-time database, etc. In this paper, the proposed method is utilized for the power ICPS. Existing datasets usually consist of only several thousand samples with a few fault categories, while in real industrial applications, users may encounter dozens of classes of possible faults. In this regard, a new large-scale dataset is collected, namely **FA**ult **R**ecognition **O**f **N**uclear power system (FARON), which contains 362,994 samples from 66 categories. Then, the effectiveness and superiority of the proposed method are verified by conducting the large-scale diagnosis task based on this new dataset and the comparison with six state-of-the-art baseline diagnosis methods in various experiments of industrial Big Data analysis, including the normal condition, and long-tailed condition that often occurs in the real industrial applications.

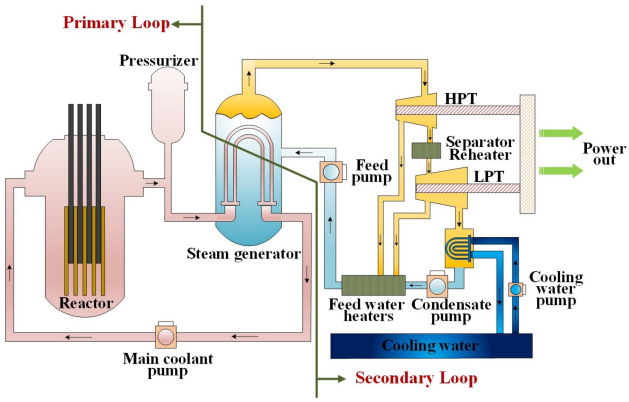


Fig. 3. Experimental Setup of the Nuclear Power System [30].

A. Data Description

The FARON dataset consists of a large number of encrypted operational data of nuclear power systems [30]. The nuclear power system is the energy generation module of a power plant, which is usually composed of steam generators, turbines, electric generators, condensers, pumps, valves, and associated equipment, as shown in Fig. 3. Normal and fault data are simulated by a nuclear power system simulator, which has 121 sensor-respond outputs for the primary system and the secondary system. During the data generation process, the nuclear power plant started from the normal state in each simulation and ran for 2 minutes; then the faults were introduced at a certain point in the operational process. Thus, the operational data of 65 fault types were collected by 19.89 h simulation, in which each fault type is simulated ranging from 10 to 77.7 minutes. To extract the spatial and temporal domain features, the time-series multi-variable samples were processed into multiple time segments. As a result, a two-dimensional $m \times n$ matrix can be obtained, where m represents the length of sampling time and n represents the number of variables. Concretely, m is set to 20, and n is set to 121. To obtain normal data, six simulations of different operational environments were performed. The sampling frequency is 4 Hz during the simulating process, and this leads to 76,632 samples under health state being collected. In the process of fault data generation, the nuclear power plant started from a normal state in each simulation, and then the faults were set to occur at a certain point in the operational process. To well describe the operational environments more comprehensively, failures of most main components of the nuclear power plant were simulated, including the feed water pump, circulating pump, condenser, and relevant valves, yielding 65 fault cases and 286,362 samples in fault data.

B. Experiments on Normal Industrial Big Data Analysis Task

1) Experimental Details: In this part, the proposed CTFN learning framework is utilized to analyze the originally collected dataset for industrial Big Data analysis of the nuclear power system. For a fair comparison, the corresponding CTFNs of

TABLE I
STRUCTURE OF IAGNN, CNN, MLP AND DBN

Layer	IAGNN model	CNN model	MLP model	DBN model
L1	GCNConv(20× 256)	Conv(1×5×32)	FC(64)	FC(64)
L2	GCNConv(256× 256)	Maxpool(1×2)	FC(128)	FC(128)
L3	GCNConv(256× 256)	Conv(1×5×64)	FC(66)	FC(66)
L4	FC(512)	Maxpool(1×2)	-	-
L5	FC(256)	FC(66)	-	-
L6	FC(66)	-	-	-

all six flat methods are set to be the same parameters. Several widely used diagnosis methods are also used to analyze the same dataset for comparison, including logistic regression (LR), principal components analysis and linear discriminant analysis (PCA+LDA), support vector machine (SVM), convolutional neural network (CNN), k-nearest neighbors (KNN), multilayer perceptron (MLP), deep belief network (DBN) and interaction-aware graph neural network (IAGNN). To test the performance of different methods on unknown detection, samples of four fault types were randomly removed from the training set and added to the test set. To minimize the effect of random fault selection, the results reported are the average of 10 trials.

In terms of unknown detection, since the performance relies on the threshold, the area under the ROC curve (AUROC) is leveraged to measure such performance in a threshold-free way. AUROC considers many possible thresholds and calculates the area as a comprehensive evaluation. Concretely, for a certain threshold, the True Positive Rate (TPR) and False Positive Rate (FPR) can be computed by $\frac{TP}{TP+FN}$ and $\frac{FP}{FP+FN}$, respectively, where TP is the true positive; FN is the false negative; FP is the false positive. By setting the TPR as the vertical axis and the FPR as the horizontal axis, the ROC curve can be plotted, and the AUROC can be computed. Following [26], 1000 samples of the training set are used to choose the parameter $\alpha_i (i = \{1, \ldots, \Phi\})$ that can obtain the best AUROC. In terms of computation efficiency, we use the testing time to compare different algorithms. Concretely, the elapsed time that an algorithm completes the target task is computed during experiments.

Implementations. The implementations by liblinear [27] are used for LR and SVM. The cost parameter is set to be 2 in LR; the components are set to be 45 in PCA+LDA and the bias is set to be 1 in both LR and SVM. The structures of CNN, MLP, DBN and IAGNN are designed by following the works in [28], [29], [30] as shown in Table I. The structures of CNN, MLP and DBN are designed by following the works in [28], [29], as shown in Table I. During training, the Adam optimization algorithm [31] is applied, where the learning rate is set as 1e-3 and other parameters are set by default. The size of a minibatch and the validation frequency are set to 200 and 30, respectively. For KNN, the number of nearest neighbors is chosen as 5. For the corresponding CTFNs, the parameters are the same as the flat ones for fair comparison. The number of outputs on each node in CNN, MLP, and DBN are set as the number of child nodes under the current node. To speed up training, principal component analysis is utilized to keep 99.5% energy. In the use of the dataset, 80% of the data is split as the training set and the rest 20% is used as the test set. Since the fault data are generated in a time sequence, it cannot be shuffled. Finally, all the results reported are the average of 10 trials.

TABLE II
ACCURACY (%, ↑), AUROC(↑), AND TESTING TIME (SECOND, ↓) COMPARISON BETWEEN LR AND SVM AND THEIR CORRESPONDING CTFNS (HL VERSIONS)

Method	LR	HL-LR	SVM	HL-SVM
Accuracy	65.87	**68.12**	54.51	**62.63**
AUROC	0.833	**0.871**	0.704	**0.745**
Testing time	0.419	0.432	0.827	0.799

TABLE III
ACCURACY (%, ↑), AUROC(↑), AND TESTING TIME (SECOND, ↓) COMPARISON BETWEEN CNN AND KNN AND THEIR CORRESPONDING CTFNS (HL VERSIONS)

Method	CNN	HL-CNN	KNN	HL-KNN
Accuracy	74.05	**78.97**	71.68	**72.69**
AUROC	0.875	**0.909**	0.813	**0.828**
Testing time	4.278	7.429	385.799	497.632

TABLE IV
ACCURACY (%, ↑), AUROC(↑), AND TESTING TIME (SECOND, ↓) COMPARISON BETWEEN MLP AND DBN AND THEIR CORRESPONDING CTFNS (HL VERSIONS)

Method	MLP	HL-MLP	DBN	HL-DBN
Accuracy	53.36	**58.49**	56.41	**60.33**
AUROC	0.608	**0.639**	0.633	**0.682**
Testing time	0.356	0.521	0.260	0.498

TABLE V
ACCURACY (%, ↑), AUROC(↑), AND TESTING TIME (SECOND, ↓) COMPARISON BETWEEN PCA+LDA AND IAGNN AND THEIR CORRESPONDING CTFNS (HL VERSIONS)

Method	PCA+LDA	HL-PCA+LDA	IAGNN	HL-IAGNN
Accuracy	66.82	**69.23**	78.83	**80.83**
AUROC	0.845	**0.885**	0.901	**0.923**
Testing time	2.326	2.036	6.673	6.762

2) Results and Analyses: Tables II, III, IV and V show the results of the classification accuracy, unknown detection AUROC, and testing time comparison between the baseline diagnosis methods and their corresponding CTFNs. It can be seen that all of the CTFNs perform better than the flat baseline methods in terms of accuracy and AUROC. For the seen industrial Big Data analysis, it can be seen that the industrial Big Data analysis task is much more difficult than the small-scale ones. The classification

TABLE VI
EXPLANATIONS OF THE FAULT CATEGORIES UNDER THE SUPERORDINATE NODE #68 IN FIG. 9

# Fault category	Explanation
19	Single auxiliary circulating pump degradation 1 (60%)
20	Single auxiliary circulating pump degradation 2 (60%)
21	Multiple auxiliary circulating pumps degradation 1 (60%)
23	Main circulating pump multi-valve failure 1
24	Steam pressure regulating valve failure
44	Single condensate pump degradation 1 (50%)
46	Multiple condensate pump degradation 1 (50%)

Number in parenthesis is the percentage of performance drop.

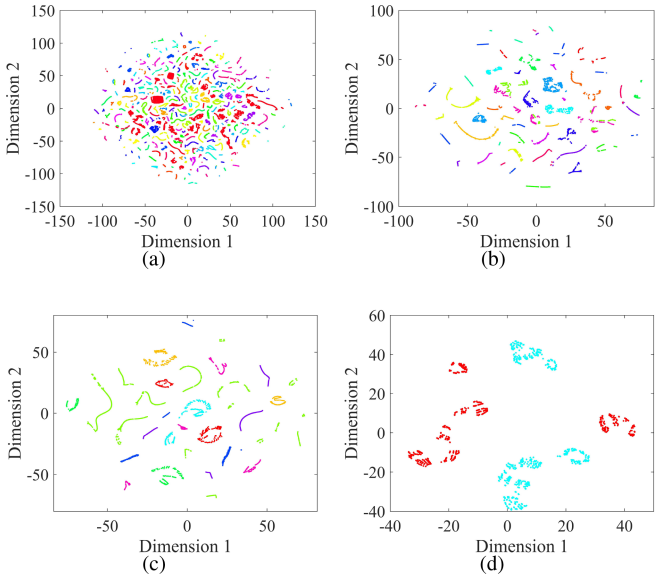


Fig. 4. Data visualization via t-SNE for (a) the original data, (b) the extracted feature map on node #68, (c) the extracted feature map on node #70, and (d) the extracted feature map on node #72. CTFN can divide a difficult problem in a complex space into several simple sub-problems on concise sub-spaces.

accuracy is generally below 80% for all the methods. The increasing number of fault categories leads to larger intra-class differences and smaller inter-class differences, which increases the difficulty in identifying all the fault categories altogether in a single-shot diagnosis. The proposed CTFN method learns to construct the CTFN by grouping similar fault categories into the same superordinate category, which enlarges the inter-class difference of the sub-problems on each of the nodes. The data distribution of the original data and the extracted feature map on some superordinate nodes of the learned structure are visualized via t-SNE [32] in Fig. 4. It can be seen that the features of the proposed CTFN are visually more discriminative than the baseline methods. Moreover, the proposed CTFN method has clear advantages over all the flat methods except KNN. This is because the complex learning problem poses great challenges to finding the optimal decision boundaries. By contrast, KNN only takes into account the information of the nearing neighbors, which alleviates the adverse effects of small inter-class differences. This might make it stable for large-scale problems, so the CTFN can only help a little for KNN by producing several simple sub-problems.

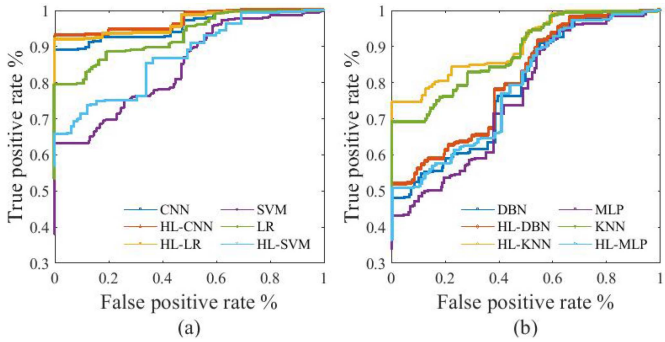


Fig. 5. Area Under ROC curve (AUROC) comparison between different methods on unknown fault detection: (a) LR, SVM, and CNN; (b) KNN, MLP, and DBN.

For detecting unknown faults, as shown in Fig. 5, the CTFN approach can take advantage of the information of multi-granularity decisions, and this benefits the task of unknown detection. Finally, the testing times of the CTFN method are shorter when using LR and SVM as base classifiers than that of flat baselines, but are longer when using other neural network-based models as base classifiers than that of flat baselines. As mentioned in Section III-C, neural-network-based methods contain a feature extraction process, so corresponding CTFNs suffer from relatively low efficiency as such a feature extraction process will be duplicated on all levels. The KNN method requires each testing sample to compute similarity with all training samples, and this also makes the HL-KNN inefficient. By contrast, for other classical classifiers, e.g., LR and SVM, the proposed methods facilitate the diagnosis efficiency through a top-down divide-and-conquer diagnosis process.

C. Experiments on Synthetic Long-Tailed Industrial Big Data

1) Experimental Details: The collected dataset contains 2,000-5,000 samples for 64 fault categories, and this is helpful in exploring rich features and differences of various faults. In fact, most fault categories are rarely seen and thus have very few samples for training, while a small proportion of fault categories usually take place in operation. This phenomenon is referred to as long-tailed distribution problem [25], or Pareto's principle. To test the performance of various methods in such real industrial scenarios, the dataset is sampled to simulate the problem of long-tailed distribution, as shown in Fig. 6.

Implementations. The first 53 fault categories (80% of the fault categories, following the 80-20 rule in Pareto's principle) are randomly sampled. An unbalanced rate is set to control the proportion of a certain category: $\hat{N}_i = N_i \times r_u$, where N_i is the number of samples in the i-th category, $\hat{N}_i$ is the number of remaining samples in the i-th category, and r_u is the unbalanced rate. Each r_u leads to new data with different degrees of long-tailed distribution. In this way, the classification accuracy can be computed at each r_u, and a curve across all r_us can be obtained. Finally, seven values of r_u are selected, i.e., $r_u = \{0.6, 0.7, 0.8, 0.85, 0.9, 0.95, 0.99\}$. To eliminate the

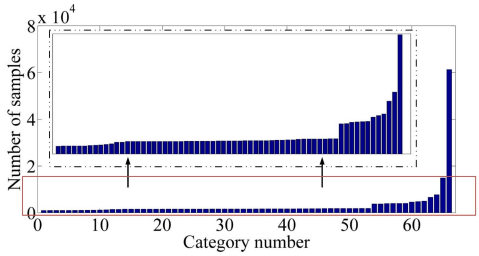


Fig. 6. Long-tailed data distribution. The original dataset is modified by randomly sampling in 80% fault categories with $r_u = 0.6$.

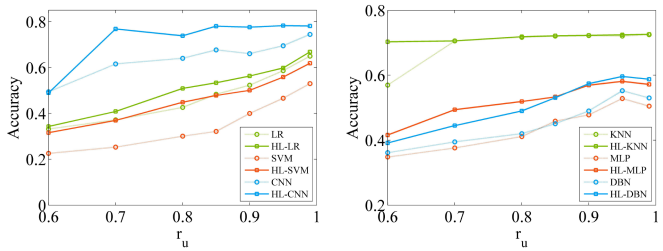


Fig. 7. Accuracy comparison of (a) LR, SVM, CNN and the corresponding CTFNs, (b) KNN, MLP, DBN and the corresponding CTFNs, with different r_us. The dashed lines in different colors represent the accuracy of the baseline methods, while the solid lines represent the accuracy of the corresponding CTFNs.

effect of random sampling, the average results of 10 trials of random sampling are reported. All the other settings are the same as Section IV-B.

2) Results and Analyses: The results are shown in Fig. 7. The accuracies of the flat methods and the corresponding CTFNs are plotted in dashed lines and solid lines respectively. It can be seen from the results that the proposed CTFNs outperform all the corresponding flat methods. Moreover, it can be indicated from the results that: (1) the accuracy is generally strongly related to the unbalanced rate, where the high rate would pose an adverse effect on the performance of the methods; (2) the tolerability to the unbalancedness varies from different method, among which CNN and KNN perform better than other methods; (3) CTFNs perform significantly better than the corresponding flat ones, and even can build the performance gap across different flat methods, e.g., the performance of HL-SVM being comparable with that of LR. It is worth noting that the HL-CNN can even eliminate the adverse effects of long-tailed distribution in some cases, e.g., the curve of HL-CNN with r_u from 0.7 to 0.99.

The reasons why CTFNs can deal with the long-tailed distribution problem better may come from two aspects. On one hand, since the CTFN is learned based on the inter-class similarity, sub-problems under some of the superordinate categories, such as node #70 and #72, are balanced. On the other hand, although data distribution is still unbalanced on other nodes, they come from fewer categories than the original problem and hence are relatively easier to handle.

D. Experiments on CTFN Architecture Learning

1) Experimental Details: To construct the CTFN architecture, a commonly used method is hierarchical spectral clustering, which requires setting two parameters: the number of child nodes per parent θ and tree depth Φ, then the best structure can be obtained by exhaustive search on the parameters. In this paper, a parameter-free hierarchical community detection method is designed to intelligently build the structure. Therefore, experiments in this part are conducted to explore and compare the performance of the proposed parameter-free method and traditional hierarchical spectral clustering.

Implementations. The range of parameters θ and Φ are selected by considering all possible values. In this experiment, the ranges that lead to good performance to report the results are selected. Finally, we set $\theta, \Phi \in \{3, 4, 5, 6, 7\}$. For a fair comparison, the inter-class similarity matrix obtained in Section IV-B is used for hierarchical spectral clustering with various parameters and the proposed parameter-free hierarchical community detection. Then, the classification accuracy is compared based on structures built by both methods. Other settings remain the same as Section IV-B and IV-C.

2) Results on CTFN Architecture Construction Algorithms: Fig. 8 displays the results of various possible combinations of parameters θ and Φ, where the dark red bar on location (8, 8) in (x, y) plane in each sub-figure is the result of the proposed parameter-free algorithm. It can be seen that the proposed parameter-free algorithm performs the best (with LR and SVM classifiers), and is close to the best (with other classifiers). This may result in some interesting conclusions in Fig. 8: (1) Neither extremely deep nor shallow hierarchies could benefit the performance of CTFN; (2) Parameter-free methods do not lose performance in all the scenarios of building the CTFN architecture. The reason may be that the optimal splits on different superordinate nodes should not always be identical (the same θ); (3) The performance varies from different classifiers with the same parameter combination of θ and Φ, because classifiers have their own biases on data, e.g., SVM preferring the cases of clear margins.

3) Analyses on the Learned CTFN Architecture: The learned CTFN architecture of fault categories (See Fig. 9) can be regarded as a kind of extracted knowledge and is a possible way to interpret the algorithm.

Some useful cues can be found to interpret some of the rationales by analyzing the learned structure. Take the superordinate node #68 as an example. Some fault categories from the same and similar components are assigned to the same group. Node #68 mainly includes faults from the circulating water pump, condenser pump, seawater pump, and main turbine governor valves. These components serve together to provide water to the pressurized water reactor for generating steam and to make the energy cycle [33], so they are highly interconnected. Nabeshima et al. pointed out that the faults of the turbine governor valve may lead to changes in steam flow and feed water flow [34]. Therefore, it is reasonable to have these components faults from these interactively related components in the same group.

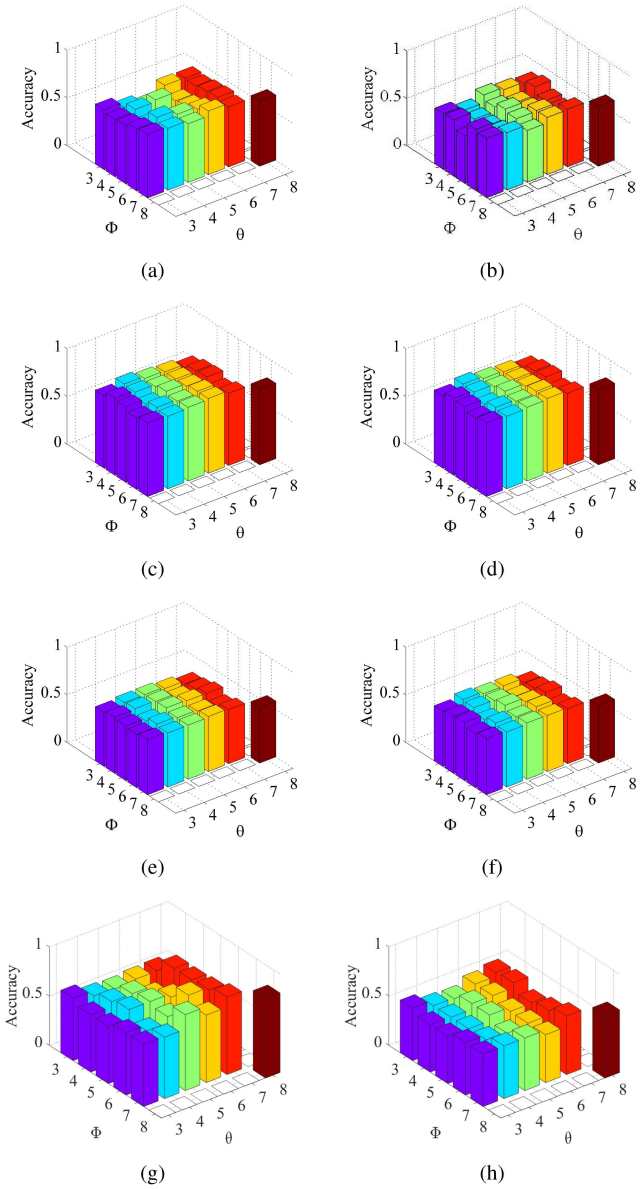


Fig. 8. Accuracy of various parameter combinations of CTFN and the proposed parameter-free hierarchical community detection with (a) LR, (b) SVM, (c) CNN, (d) KNN, (e) MLP, (f) DBN, (g) PCA+LDA and (h) IAGNN. Dark red bars at location (8, 8) of (x, y) plane show accuracies of the parameter-free hierarchical community detection, while other bars represent the accuracy of different parameter combinations of CTFN.

Moreover, better performance can be obtained by grouping these fault categories. Fig. 10 visually shows the confusion matrix of CTFN under node #68 and the same part of the confusion matrix of flat methods (both using the CNN classifier). It can be found that the flat methods consider all faults at the same time and misclassify a large proportion of samples. By contrast, the proposed CTFNs clearly improve the diagnosis performance by grouping these fault categories together and identifying them without interference from other fault categories. Fig. 10 shows the results of industrial Big Data analysis in terms of the confusion matrix, in which the diagonal elements are the correct predictions while other positions depict the number of misclassified samples. To visualize the results, bright colors

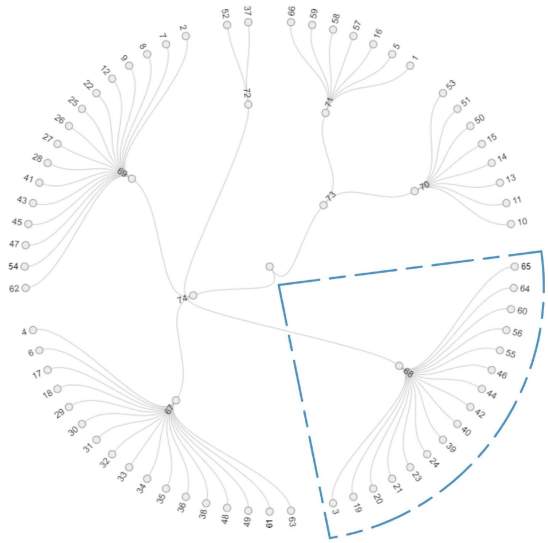


Fig. 9. Visualization of the learned CTFN architecture of fault categories. Detailed explanations of the fault categories under the superordinate node #68 are shown in Table VI.

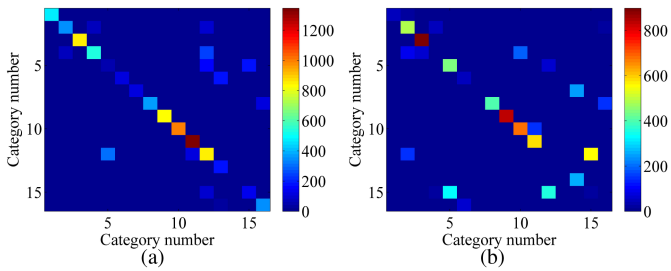


Fig. 10. Confusion matrix comparison under node #68 of (a) CTFN, and the same part of (b) flat method (CNN classifier) for fault categories in Fig. 9.

represent a larger number of samples while dark colors represent a less number of samples. It can be seen that the proposed method performs better than the baseline methods, which verifies its effectiveness.

V. Conclusion

In this paper, an industrial Big Data analytical system is proposed for industrial Big Data analysis and condition monitoring of complex systems. Experimental results illustrated that the proposed Big Data analytical system has clear advantages over other baseline flat methods. Though the proposed system performs well on dozens of known industrial Big Data analyses, there are still many challenges in real engineering that have not been solved or considered in this paper, such as the data coupling and analysis problems. We will investigate the intelligent equipment design or improvement based on the proposed industrial Big Data analytical system, and look forward to constructing an integrated and finer ICPS system. In real-world applications, the proposed method can be trained distributively with popular techniques such as SPARK to efficiently handle large-scale tasks. In the future, we will improve the recognition capability

of new fault classes, thus improving the practicability of the proposed method in real engineering.

REFERENCES

[1] T. Chen, D. J. Hill, and C. Wang, "Distributed fast fault diagnosis for multimachine power systems via deterministic learning," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4152–4162, May 2020.

[2] P. Hu and J. Zhang, "5G-enabled fault detection and diagnostics: How do we achieve efficiency?," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3267–3281, Apr. 2020.

[3] K. Zhu, G. Li, and Y. Zhang, "Big data oriented smart tool condition monitoring system," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4007–4016, Jun. 2020.

[4] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, 2018.

[5] J. Yu and X. Zhou, "One-dimension residual convolutional auto-encoder-based feature learning for gearbox fault diagnosis," *IEEE Trans. Ind. Inform.*, vol. 16, no. 10, pp. 6347–6358, Oct. 2020.

[6] S. Zhang, Z. Sun, M. Wang, J. Long, Y. Bai, and C. Li, "Deep fuzzy echo state networks for machinery fault diagnosis," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1205–1218, Jul. 2020.

[7] S. Xing, Y. Lei, S. Wang, and F. Jia, "Distribution-invariant deep belief network for intelligent fault diagnosis of machines under new working conditions," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2617–2625, Mar. 2021.

[8] R. Liu, B. Yang, and A. G. Hauptmann, "Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network," *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 87–96, Jan. 2020.

[9] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, Jul./Sep. 2020.

[10] Z. Wu, H. Jiang, K. Zhao, and X. Li, "An adaptive deep transfer learning method for bearing fault diagnosis," *Measurement*, vol. 151, 2020, Art. no. 107227.

[11] F. Wang, R. Liu, Q. Hu, and X. Chen, "Cascade convolutional neural network with progressive optimization for motor fault diagnosis under non-stationary conditions," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2511–2521, Apr. 2021.

[12] D. Xiao, J. Ding, X. Li, and L. Huang, "Gear fault diagnosis based on kurtosis criterion VMD and SOM neural network," *Appl. Sci.*, vol. 9, no. 24, 2019, Art. no. 5424.

[13] P. Liu, Y. F. Zhang, H. Wu, and T. Fu, "Optimization of edge-PLC based fault diagnosis with random forest in industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9664–9674, Oct. 2020.

[14] T. Zhao et al., "Embedding visual hierarchy with deep networks for large-scale visual recognition," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4740–4755, Oct. 2018.

[15] D. Erhan, P. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," *J. Mach. Learn. Res.*, vol. 5, pp. 153–160, 2009.

[16] H. Huang et al., "Real-time fault-detection for IIoT facilities using GBRBM-based DNN," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5713–5722, Jul. 2020.

[17] J. C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining Knowl. Discov.*, vol. 22, no. 1–2, pp. 31–72, 2011.

[18] K. Aris, P. Ioannis, G. Eric, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: A unified view and novel approaches," *Data Mining Knowl. Discov.*, vol. 29, no. 3, pp. 820–865, 2015.

[19] J. Fan, N. Zhou, J. Peng, and L. Gao, "Hierarchical learning of tree classifiers for large-scale plant species identification," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4172–4184, Nov. 2015.

[20] Y. Qu et al., "Joint hierarchical category structure learning and large-scale image classification," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4331–4346, Sep. 2017.

[21] F. R. Bach and M. I. Jordan, "Learning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 305–312.

[22] Y. Wang et al., "Deep fuzzy tree for large-scale hierarchical visual classification," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1395–1406, Jul. 2020.

[23] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mechanics*, vol. 2008, no. 10, pp. 155–168, 2008.

[24] D. Koller and M. Sahami, "*Hierarchically Classifying Documents Using Very Few Words*," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., 1997.

[25] Y. Zhou, Q. Hu, and Y. Wang, "Deep super-class learning for long-tail distributed image classification," *Pattern Recognit.*, vol. 80, pp. 118–128, 2018.

[26] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

[27] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[28] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, Apr. 2019.

[29] R. Liu, F. Wang, B. Yang, and S. J. Qin, "Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 3797–3806, Jun. 2020.

[30] D. Chen, R. Liu, Q. Hu, and S. X. Ding, "Interaction-aware graph neural networks for fault diagnosis of complex industrial processes," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, vol. 34, no. 9, pp. 6015–6028, Sep. 2023.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[32] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[33] A. Lancha, M. Serrano, J. Lapena, and D. Gómez-Briceño, "Failure analysis of a river water circulating pump shaft from a NPP," *Eng. Failure Anal.*, vol. 8, no. 3, pp. 271–291, 2001.

[34] K. Nabeshima et al., "On-line neuro-expert monitoring system for borssele nuclear power plant," *Prog. Nucl. Energy*, vol. 43, no. 1-4, pp. 397–404, 2003.

Ruonan Liu (Member, IEEE) received the B.S., M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively. She was a postdoctoral Researcher with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, in 2019. She is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include machine learning, intelligent manufacturing and computer vision.

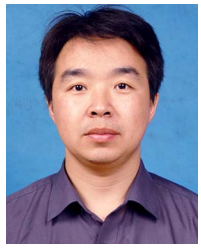
Quanhu Zhang received the B.S. degree in computer science and technology in 2022 from Nanjing Forestry University, Nanjing, China, where he is currently working toward the M.S. degree in electronic Information engineering with the College of Intelligence and Computing. His research interests include deep learning and fault diagnosis.

Yu Wang (Member, IEEE) received the B.S. degree in communication engineering, the M.S. degree in software engineering, and the Ph.D. degree in computer applications and techniques from Tianjin University, Tianjin, China, in 2013 and 2016, and 2020, respectively. He is currently an Assistant Professor of Tianjin University. His research interests include hierarchical learning and large-scale classification in industrial scenarios and computer vision applications, data mining and machine learning.

Zengxiang Li received the B.S. degree from the Shanghai University of Electric Power, Shanghai, China, in 2003, the M.S. degree from Shanghai Jiao Tong University Shanghai, in 2006, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2012 respectively. He was a Research Scientist of High Performance Computing, A*STAR, Singapore, from 2012 to 2020. He is currently the Executive Vice President of Digital Research Institute of ENN Group. He has authored or coauthored more than 50 high-quality papers on ACM/IEEE Transactions, reputable journals and conferences. His research interests include, distributed system, graph computing, industrial IoT, blockchain, AI, federated learning, incentive mechanism, and privacy-preserving technology.

Dongyue Chen received the B.S. degree in transportation equipment information engineering, and the M.S. degree in precision instruments and machinery from Southwest Jiaotong University, Chengdu, China, in 2015 and 2018, respectively. She is currently working toward the Ph.D. degree in computer applications and techniques with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include deep learning and condition monitoring of mechanical system.

Steven X. Ding received the Ph.D. degree in electrical engineering from Gerhard–Mercator University, Duisburg, Germany, in 1992. From 1992 to 1994, he was a Research and Development Engineer with Rheinmetall GmbH, Germany. From 1995 to 2001, he was a Professor of control engineering with the University of Applied Science Lausitz, Senftenberg, Germany, and the Vice President from 1998 to 2000. He is currently a Full Professor of control engineering and the Head of the Institute for Automatic Control and Complex Systems (AKS) with the University of Duisburg-Essen, Germany. His research interests include model-based and data-driven fault diagnosis, fault tolerant systems, real-time control, and their application in industry with a focus on automotive systems, chemical processes, and renewable energy systems.



Qinghua Hu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. From 2009 to 2011, he was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University,, Hong Kong. He is currently the Dean of the School of Artificial Intelligence, Vice Chairman of the Tianjin Branch of China Computer Federation, Vice Director of the SIG Granular Computing and Knowledge Discovery, and Chinese Association of Artificial Intelligence. He is currently supported by the Key Program, National Natural Science Foundation of China. He has authored or coauthored more than 200 peer-reviewed papers. His research interests include uncertainty modeling in Big Data, machine learning with multi-modality data, intelligent unmanned systems. He is an Associate Editor for IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Acta Automatica Sinica*, and *Energies*.

Boyuan Yang (Member, IEEE) received the B.A., M.A., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively. He was a Research Associate with the School of Electrical and Electronic Engineering, University of Manchester, Manchester, U.K. In 2020, he joined the College of Artificial Intellegence, Nankai University, as an Associate Professor. His research interests include intelligent manufacturing, machine learning, condition monitoring, and wind energy.