Pattern Recognition 123 (2022) 108383

Contents lists available at ScienceDirect

Pattern Recognition

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

Majorities help minorities: Hierarchical structure guided transfer learning for few-shot fault recognition

Hao Chen[a,b], Ruonan Liu[a,*], Zongxia Xie[a], Qinghua Hu[a], Jianhua Dai[c], Junhai Zhai[b]

[a] College of Intelligence and Computing, Tianjin University, Tianjin, China
[b] Hebei Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding 071002, Hebei, China
[c] Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, China

ARTICLE INFO

Article history:
Received 22 April 2021
Revised 22 September 2021
Accepted 19 October 2021
Available online 24 October 2021

Keywords:
Transfer learning
Fault recognition
Few-shot problem
Hierarchical category structure
Complex systems

ABSTRACT

To ensure the operational safety and reliability, fault recognition of complex systems is becoming an essential process in industrial systems. However, the existing recognition methods mainly focus on common faults with enough data, which ignore that many faults are lack of samples in engineering practice. Transfer learning can be helpful, but irrelevant knowledge transfer can cause performance degradation, especially in complex systems. To address the above problem, a hierarchy guided transfer learning framework (HGTL) is proposed in this paper for fault recognition with few-shot samples. Firstly, we fuse domain knowledge, label semantics and inter-class distance to calculate the affinity between categories, based on which a category hierarchical tree is constructed by hierarchical clustering. Then, guided by the hierarchical structure, the samples in most similar majority classes are selected from the source domain to pre-train the hierarchical feature learning network (HFN) and extract the transferable fault information. For the fault knowledge extracted from the child nodes of one parent node are similar and can be transferred with each other, so the trained HFN can extract better features of few samples classes with the help of the information from similar faults, and used to address few-shot fault recognition problems. Finally, a dataset of a nuclear power system with 65 categories and the widely used Tennessee Eastman dataset are analyzed respectively via the proposed method, as well as state-of-the-art recognition methods for comparison. The experimental results demonstrate the effectiveness and superiority of the proposed method in fault recognition with few-shot problem.

1. Introduction

With the increased demand of functionality, quality and service, industrial systems have been developed or integrated more and more complex [1]. To ensure the safety operation and economic benefits, fault recognition of complex systems is becoming an essential task in modern industry [2].

In literature, because shallow learning models are unable to extract complex features, classical fault recognition methods usually extract fault features via signal processing techniques firstly [3], including Fourier transform, wavelet transform, empirical mode decomposition and sparse representations; then classify the fault types via artificial intelligent (AI) methods, such as support vector machine (SVM) [4], deep neural network (DNN) [5], and so on. However, for many tasks, many factors of variation that can explain the observation data affect each data at the same time, so it is difficult to know how to extract high-level abstract features as the input of AI methods. Aiming at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features, deep learning methods show the potential to overcome the aforementioned deficiency in current intelligent fault recognition methods and have been applied for fault recognition successfully [6]. In addition, with the development of sensor and communication techniques, industrial data has been collected and accumulated rapidly, which also makes it possible to train deep learning models via these big data [7]. Autoencoder [8], recurrent neural network (RNN) [9] and convolutional neural network (CNN) [10] are the most commonly applied deep learning methods, that outperform the conventional model-based approaches using large amounts of training data. However, in reality, because the equipment is generally in normal operation state for a long time, there are many normal state data obtained, while the failure state data is few, resulting in high repetition of informa-

* Corresponding author.
E-mail addresses: chenhaohbu@outlook.com (H. Chen), ruonan.liu@tju.edu.cn (R. Liu), caddiexie@hotmail.com (Z. Xie), huqinghua@tju.edu.cn (Q. Hu), jhdai@hunnu.edu.cn (J. Dai), mczjh@126.com (J. Zhai).

https://doi.org/10.1016/j.patcog.2021.108383
0031-3203/

tion in the data and lack of typical fault information. And although a large amount of monitoring data has been accumulated, only a few of the data corresponding to the health status are known, while the manual labeling data is time-consuming and costly. The scarcity of labeled data makes it difficult to train and obtain high-precision intelligent diagnostic models.

To learn and transfer the fault knowledge from the supervised training domain to the unsupervised testing domain, transfer learning has been proposed [11]. In general, transfer learning methods can be divided into three categories: instance-based, feature-based and model-based transfer learning [12]. Among them, maximum mean discrepancy (MMD) and domain adaptation have been most commonly applied for fault recognition, which belong to the feature-based transfer learning methods and provide a solution to the problem of insufficient labeled samples in engineering scenarios. For example, Yang et al. introduced MMD to bearing fault diagnosis to transfer the fault information of bearings from laboratory bearings to locomotive bearings [13]. Liu et al. proposed a deep adversarial domain adaptation model based on a deep stack autoencoder for bearing fault diagnosis, which has been verified through six domain adaptation situation studies [14]. Wang et al. proposed a linear discriminant analysis (LDA)-based transfer learning method for fault classification of chemical processes, in which a weighted MMD is designed for domain adaptation [15]. Motivated by the domain adaptation, Li et al. proposed a deep adversarial transfer learning network for machinery emerging fault detection, which only accomplishes the discrimination knowledge transfer from the source domain to the target domain, but also implements the detection for new emerging fault class in the target domain [16]. Qian et al. proposed a deep transfer network (DTN) based on weighted joint distribution alignment (WJDA) for rotating machine fault diagnosis with working condition variation [17]. Afridi et al. proposed an automatic source selection algorithm for transfer learning in convolutional neural networks [18]. However, the above methods assume that the target domain samples are sufficient.

Aiming at the problem of few-shot learning, some methods have been proposed. Li et al. proposed an algorithm called adaptive hyper-ring detector (AHr-detector) to anomaly detection and fault diagnosis with online adaptive learning under small training samples, which has both classification and clustering functions [19]. Wu et al. proposed few-shot transfer learning method is constructed utilizing meta-learning for few samples diagnosis in variable conditions [20]. Qu et al. pointed out that hierarchical visual data structures can help for improving the efficiency and performance of large-scale multi-class classification [21]. Tai et al. proposed that the missing information across classes can be compensated by using side information [22]. Li et al. proposed a large-scale FSL model by learning transferable visual features with the class hierarchy which encodes the semantic relations between source and target classes [23].

From the literature, it can be found that the fault recognition of a single component marked by distinct characteristics has received mass concern, and the related research achievements are remarkable [24]. However, few researches focus on fault recognition of complex systems, which is meeting different challenges. Firstly, the complex physical structure of these systems will lead to numerous fault classes. And these failure data usually cannot keep balance in real engineering and thereby show long-tail distributions. That is, most failure data distributed in few common fault classes, and the available data of most fault classes is small, which will increase the modeling difficult of these rare faults with few-shot samples. Transfer learning is simple and straight forward, but cannot be implemented when the data of target domain is small or limited. Secondly, although both the dataset and fault classes have been increased, not all collected data is useful to transfer. For example, the key information for distributed fault recognition is the global feature, such as the vibration amplitude; while for partial fault recognition, the impact components are more important. In this case, the knowledge of distributed faults cannot help, and the model will pay too much attention to irrelevant fault features and reduce the performance of target fault recognition. This irrelevant interference problem, or so-called negative optimization problem, is less obvious in traditional fault recognition task with enough training and test data, but will become more and more serious in few-shot fault recognition of complex systems with the increase of fault classes.

To overcome the above problems, a transfer learning framework guided by a hierarchical category structure (HGTL) is proposed in this paper. Firstly, fusing domain knowledge, label semantics and inter-class distance to calculate the affinity between categories, the fault category hierarchical tree is constructed via hierarchical clustering. Then, the similar faults with the both source domain and target domain can be clustered together and used for pre-training a HFN and extract transferable fault features by the hierarchical information of fault categories. Finally, the fault feature vectors of the target class support samples are used to train an appropriate classifier for multi class classification of unknown fault samples in the target domain.

The primary contributions of this paper can be summarized as follows.

1. A novel hierarchical structure guided transfer learning framework (HGTL) is proposed for fault recognition with few-shot samples in complex systems. The fault knowledge extracted from some recognition tasks can be shared and transferred to similar fault recognition task via transfer learning techniques.
2. To filter out irrelevant recognition tasks and retain useful tasks for knowledge transfer, hierarchical category structure is used to guide the learning of more relevant cross-domain information. Therefore, a recognition model can be shared across domains and overcome the problem of negative transfer.
3. A condition dataset of a nuclear power system with 65 fault categories, as well as the Tennessee-Eastman (TE) process dataset are analyzed to verify the effectiveness and the robustness of the proposed method. The experimental results demonstrated the proposed method can be a promising tool for few-shot fault recognition of complex systems.

The rest of this paper is organized as follows. Section 2 describes the proposed method in detail. Experimental verifications of a nuclear power system and the TE process dataset are conducted in Section 3 and Section 4, respectively. Finally, Section 5 concludes this paper.

2. Proposed approach

Firstly, the few-shot fault recognition problem is formally defined as follows. Let $D_{source} = \bigcup C_i^s$ and $D_{target} = \bigcup C_i^t$ denote the source domain and target domain, respectively, where C_i^s is the ith source class and C_i^t is the ith target class. These two domains are disjoint, i.e., $D_{source} \cap D_{target} = \emptyset$. There are sufficient labeled samples in each source domain class, but only a few (less than 25 in this paper) labeled samples in each target domain class. Let S_{source} denote the sample set for D_{source}, $S_{support}$ and S_{test} denote the training and test sample sets for D_{target}, respectively. The goal of the proposed approach is to achieve good classification performance on the S_{test}.

2.1. Hierarchical category structure

Secondly, a hierarchical tree structure of fault categories is constructed to encode the affinity between all fault categories in

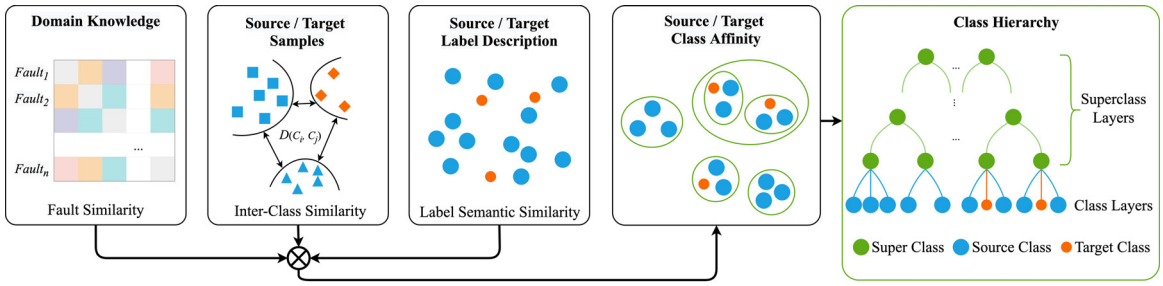


Fig. 1. Illustration of tree-structured class hierarchy construction by fusing domain knowledge, label semantics and inter-class distance. Note that the short fault description text is used as the class label.

source and target domain. The affinity measure of multi-source fusion and the establishment of hierarchical structure, as shown in Fig. 1. By fusing domain knowledge and similarity information between classes in a data-driven way, the fault similarity space is constructed and hierarchical clustering is carried out.

2.1.1. Manual measurement of fault similarity

Specifically, with the help of the prior domain knowledge of experts, the fault affinity is preliminarily identified and measured, then a fault similarity matrix A^K is obtained. Its rows and columns correspond to all source classes and target classes. The similarity value between any two faults in A^K is given manually according to experience.

2.1.2. Similarity measurement in sample feature space

Inspired by the work of Qu et al. [21], we use an inter-class distance measurement considering intra-class spread information to measure the similarity between few sample classes in the target domain and rich sample classes in the source domain.

Suppose that there are N_i samples in the ith fault C_i, where $C_i \in D_{source}$ or $C_i \in D_{target}$, and each sample represented by the features $\{x_l^i\}_{l=1}^{N_i}$. The distance between each two categories can be formulated as,

$$D(C_i, C_j) = \sqrt{\left\| Q_i - Q_j \right\|^2 + \sigma_i^2 + \sigma_j^2}, \tag{1}$$

where $Q_i = \frac{1}{N_i}\sum_{l=1}^{N_i} \boldsymbol{x}_l^i$ is the mean vector of the ith category and $\sigma_i^2 = \sum_{l=1}^{N_i} \frac{1}{N_i}\left(Q_i - \boldsymbol{x}_l^i\right)^2$ is the variance of the ith category.

The inter-class similarity matrix A^S on the sample original feature space is computed as,

$$A_{ij}^S = \exp(-\frac{D(C_i, C_j)}{\delta_{ij}}), \tag{2}$$

where δ_{ij} is the scaling factor for the similarity calculation in [21].

2.1.3. Similarity measurement in label semantic space

The cosine distance is used to measure the semantic similarity, and the inter-class similarity matrix A^L in the semantic space is calculated by,

$$A_{ij}^L = \cos(\vec{d}_i, \vec{d}_j), \tag{3}$$

where $\vec{d}_i$ is the document vector obtained by a gensim-based Doc2vec model, which is trained on fault description text.

2.1.4. Fusion into an affinity space

Using function F_{use} to fuse the above three matrices, a unified affinity metric for all categories in the source / target domain is obtained. Different data sets can flexibly select the fusion function (such as average, maximum, etc.) according to the actual situation. It is formalized as,

$$A_{ij} = F_{use}(A^K, A^S, A^L, \vec{w}), \tag{4}$$

where $\vec{w}$ weights each similar matrix. The value of $\vec{w}$ is set by manual experience and experiment in this paper. It can also be obtained by some learning algorithm [22]. Different values will change the measurement results of the distance between classes, resulting in different class hierarchies. Therefore, the bad $\vec{w}$ values will degrade the final performance of our method.

2.1.5. Hierarchical clustering on fusion affinity metric space

We adopt the bottom-up strategy to build a hierarchy of categories. Both the source class and the target class are exploited as leaves of the tree, constituting the bottom layer of the hierarchy. According to the affinity matrix, starting from the fault class corresponding to the leaf node, clustering the affinity vector of the lower node, each cluster corresponds to an upper parent node (i.e., the superclass node), and its affinity vector is the average of its subclasses. Superclasses at the same level can be clustered again until there is only one superclass, and the final superclass corresponds to the root node of the entire tree. The tree is composed of one class layer and n superclass layers, as shown in Fig. 1. Generally, each tree node maps a class set, which is formalized as,

$$\begin{cases} S_v^l &= \bigcup_{i \in S_v^l} S_i^{l-1} \quad , \quad l \in \{1, \ldots, n\} \\ S_v^0 &= \{C_v\} \quad , \quad v \in \{1, \ldots, m\} \end{cases}, \tag{5}$$

where $m = |D_{source}| + |D_{target}|$ is the number of all classes, and $S_i^l \cap S_j^l = \emptyset, i \neq j$. To simplify, let l_0 and $l_i (i = 1, \ldots, n)$ denote the class layer and the nth superclass layers, respectively. Similar classes are obviously clustered into a group. Because superclasses share information between source and target classes, the hierarchy can guide us to learn better features from the samples of the source classes that are helpful to identify the target class.

2.2. A hierarchical structure guided transfer learning framework

Fig. 2 shows the pipeline for our proposed hierarchy-guided transfer learning framework. Given a target multi-classification task with few labeled samples, we perform the following.

2.2.1. Selection of source classes guided by hierarchy

With the help of the hierarchical affinity structure covering the source classes and the target classes, the source classes are filtered first. We only select the samples in the source classes that are most relevant to the target classes (i.e., Fig. 2 ①) for transferability feature learning. The sample of the source class is usually sufficient, and the number of categories is often larger than that in the target domain. According to the hierarchical structure of fault categories, the fault feature of the source domain class and the target class within the same superclass set are similar and easier to transfer; while the source domain and the target class belonging to different superclasses are more difficult to obtain effective transferable fault features, even lead to negative transfer and reducing the classification performance of the target class. For example, cat and tiger

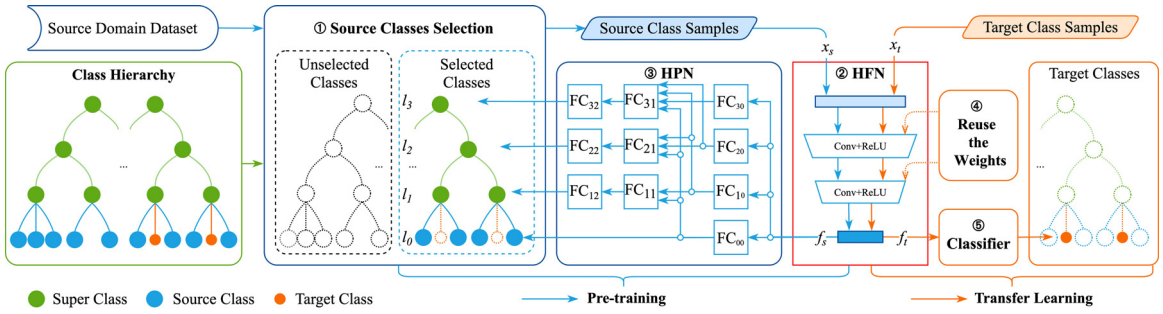


Fig. 2. Overview of the proposed hierarchy guided transfer learning framework. In this example, a 3 superclass layers hierarchy is constructed firstly to encode relations between source and target classes. Secondly, with the help of HFN and HPN, transferable features are learned by mining prior knowledge in class hierarchy. The blue box represents the pre-training process with the selected classes in source domain, and the orange box represents the classification process with few samples in target domain. Finally, the weight parameters of HFN obtained by the pre-training process are transferred to the target domain classification task. Notation: 'FC'— fully-connected network, 'x_s'— a sample in selected source class, 'x_t'— a sample in target class, 'f_s'— feature of x_s, 'f_t'— feature of x_t. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

within the same superclass have more similar feature than cats and birds that within different superclasses. In addition, the more the source domain categories are, the more complex the model needs, which will lead to a more difficult pre-training, and a worse performance of feature extraction. Therefore, guiding by the hierarchical structure of classes, more targeted selective pre-training can not only reduce the possibility of negative transfer, but also extract effective features efficiently. The source class selection guided by hierarchy is presented in Algorithm 1 .

Algorithm 1: Selection of source classes guided by hierarchy.

Input: Hierarchical structure T, Target classes set D_{target}, Superclass level of screening l
Output: Selected source classes $\hat{D}_{source}$

$\hat{D}_{source} = \emptyset$;
foreach *superclass* v *such that* $Level(v) = l$ **do**
  $S_v = \bigcup C_i$ when C_i is leaf node of a subtree with root node v;
  foreach C_i^t *in* D_{target} **do**
    if $C_i^t \in S_v$ **then**
      $\hat{D}_{source} \leftarrow \hat{D}_{source} \cup S_v$;
$\hat{D}_{source} \leftarrow \hat{D}_{source} - D_{target}$ // Exclude all target classes;
Return $\hat{D}_{source}$;

The parameter l can be used to control the filtering granularity. When $l = 1$, source classes are selected according to the highest similarity, while when $l = Level(root)$, all source classes are used for pre-training.

2.2.2. Transferable feature learning

After filter out the unrelated faults, the HFN is pre-trained by using these highly similar source classes and their hierarchical relationship with the target classes (i.e., Fig. 2 ②). HFN attempts to learn an ability to extract cross-domain feature representation, generally using a multi-layer convolution neural network structure, and this ability will be directly transferred to the target domain to extract target sample features. Therefore, the HFN can learn the prior knowledge integrated in the class hierarchy and extract the transferable fault features, which can effectively help the fault recognition task in the target domain.

Inspired by the work of Li et al. [23], we construct a hierarchical prediction network (HPN) as shown in Fig. 2 ③. HPN uses features extracted by HFN to predict class / superclass labels of each layer at the same time, which constrains the HFN to learn transferable features that are more suitable for representing the target classes. In addition, HPN can encode the hierarchical structure of the class / superclass layer. Specifically, we combine the prediction results of a certain superclass layer and its lower layers to infer the superclass label of the layer. Since the hierarchical structure between adjacent layers is shared and transmitted between the source class and the target class, hierarchical coding can further improve the transferability of the learned features.

Taking a 4-layer HPN network in Fig. 2 as an example, a fully-connected network module (FC_{00}) with softmax layer is added after the HFN model for the lowest class layer (i.e., layer l_0). Given a sample, FC_{00} module can predict the probability distribution of classes. To model the hierarchical structure between adjacent layers, for the lowest superclass layer (i.e., layer l_1), the outputs of FC_{00} in layer l_0 and the outputs of FC_{10} in layer l_1 are combined as input of the subsequent FC_{11}. Then, the FC_{12} module with softmax layer outputs the final superclass prediction results of layer l_1. Its formal formulation is given as:

$$\hat{p}_{l_1} = F_{l_1}^2(p_{l_0} \oplus p_{l_1}), \tag{6}$$

where p_{l_0} denotes the output of the bottom FC_{00} module which means the prediction results of class layer l_0, p_{l_1} denotes the output of FC_{10} module which means the prediction results of the lowest superclass layer l_1 in first step. $\oplus(.)$ is a concatenation operator by channel, and $F_{l_1}^2(.)$ is the second step prediction composed of FC_{11} and FC_{12} modules corresponding to layer l_1. The output $\hat{p}_{l_1}$ denotes a final predicted distribution over all possible superclass labels at the layer l_1 of the hierarchy.

Similarly, for any superclass layer $l_i (i = 1, \ldots, n)$, the superclass labels can also be inferred by combining the first outputs of the layers $\{l_j : j \le i\}$ as its input. Its general formal formulation is:

$$p_{l_i} = F_{l_i}^1(G(x)), (i = 0, \ldots, n), \tag{7}$$

$$\hat{p}_{l_i} = F_{l_i}^2(\oplus_{j=0}^{i} p_{l_j}), (i = 1, \ldots, n), \tag{8}$$

where G denotes a forward step of the HFN for feature extraction. $F_{l_i}^1$ and $F_{l_i}^2$ respectively denote a forward step of the FC network corresponding to layer l_i in the first and second step. p_{l_i} denotes the predicted distribution over possible classes/superclasses in layer $l_i (i = 0, \ldots, n)$ in the first step. $\hat{p}_{l_i}$ denotes the final predicted distribution over possible superclasses in layer $l_i (i = 1, \ldots, n)$.

Therefore, by combining the prediction results of all class/superclass layers, the loss function of sample x can be defined as follows:

$$Loss(x, Y; \Theta) = \lambda_0 L(y_{l_0}, p_{l_0}) + \sum_{i=1}^{n} \lambda_i L(y_{l_i}, \hat{p}_{l_i}), \sum_{i=0}^{n} \lambda_i = 1, \tag{9}$$

where $Y = \{y_{l_i}, i = 0, \ldots, n\}$ collects the true class/super-class labels of sample x, where y_{l_i} denotes the label corresponding to layer l_i. Θ denotes the parameters of the full network. L denotes the cross-entropy loss for classification, and λ_i weights these losses.

2.2.3. Target classification prediction under few-shot

Once HFN is trained with the selected source class data, it can be used to extract transferable features for fault samples from target classes (i.e., $S_{support}$ and S_{test}), as shown in Fig. 2 ④. With these transferable fault features, some simple classifier models (i.e., Logistic Regression, Nearest Neighbors Search, Nearest Centroid, Linear SVM) can be used to infer the labels of test samples in S_{test}, as shown in Fig. 2 ⑤.

3. Case study 1

3.1. Experimental setup

3.1.1. Data description

In this paper, a new dataset is collected, namely FAult Recognition of Nuclear power system (FARON). The FARON dataset consists of many encrypted monitoring data of nuclear power system, collected from major components, including feedwater pump, circulating pump, condenser and related valves. The different working conditions of the primary system and the secondary system were simulated by using a nuclear power system simulator with 121 sensor response outputs. During the simulation, the sampling frequency was 1 Hz, and 58,829 samples with Gaussian noise under 65 different health conditions were collected. Tables 1 and 2 list the sensor types and fault types respectively.

The data set is divided into two domains. The source domain contains 55 categories of faults, with a total of 50,273 samples, of which the largest class has 2204 samples, and the smallest class has 478 samples. 60% of the samples from each fault are randomly selected to form the training set, and the rest of the samples are used as the verification set. The target domain consists 10 categories of faults, with a total of 8556 samples, of which the largest class has 965 samples, and the smallest class has 657 samples. Then n samples ($n \leq 25$) from each fault are randomly selected to form the support set, and the rest are used as the test set.

3.1.2. Hierarchical fault category structure

We use the JIEBA Chinese word segmentation toolkit to segment the description text of each fault, and use the doc2vec model of the GENSIM toolkit to generate the semantic vector of the fault label. In the experiment, each label text is expressed as a 100-dimensional semantic vector, and the cosine distance is used to measure the similarity. Using domain knowledge, we manually measure the similarity between faults. The main heuristic rules include (1) the similarity of failures on the same component is higher (2) the similarity of the failures of structurally related components is higher. For example, the multi-valve failure of the exhaust pressure control valve is very similar to the multi-valve failure of the inlet valve of main circulating water pump. In this work, when the fault categories are leaf nodes, and the number of nodes in each layer is 65, 32, 10 and 4, then a 3 superclass layers hierarchy is constructed. The fault category hierarchy used in the experiment is shown in Fig. 3.

3.1.3. Pre-training details of source domain

As shown in Fig. 3, when $l = 1$, we select samples of 15 types of faults (blue nodes) under the superclass (green nodes) of the target domain class (orange node) as the source domain training set to pre-train the HFN, and discard samples of other source domain classes (gray nodes). The HFN is mainly composed of two layers

Table 1
Sensor types on FARON dataset.

Type	Sensor	Type	Sensor
Atmospheric relief flow	21, 22	Main steam flow	17, 18
Circulating pump inlet valve opening	58, 59	Intake regulate valve opening	31, 32
Condensate pump pressure	10, 46, 47, 48, 49, 73, 74, 75, 76	Valve pressure	13, 83, 84, 85, 86, 87, 88, 89, 90, 91
Condensate temperature	60, 61, 62, 63	Header pressure	29, 70
Condenser pressure	43, 64, 65	Header temperature	67, 66
Condenser temperature	38, 40, 41, 42	Hot well temperature	39
Coolant flow	112, 113	Outlet steam temperature	4, 5
Coolant pressure	104, 105, 106, 107	Power	1
Coolant temperature	95, 96, 97, 98	Pressurizer pressure	101, 102, 103
Cooling water flow	30, 50, 51	Pressurizer temperature	100
Cylinder inlet pressure	93	Rod position	118, 119, 120, 121
Differential pressure	14, 15, 16, 44, 45, 71, 72, 108, 109, 110, 111	Speed	25, 26, 54, 55, 79, 80, 92, 114, 115, 116, 117
Feedwater flow	19, 20	Steam pressure	33, 34, 35, 36, 37, 68
Feedwater heater temperature	2, 3	Vacuum pressure	69, 94
Feedwater pump pressure	6, 7, 8, 9, 11, 12	Water level	23, 24, 52, 53, 77, 78, 99
Feedwater regulate valve opening	27, 28	Water level regulating valve opening	56, 57, 81, 82

Table 2
Fault types on FARON dataset.

Type	Fault
Air extractor failure	54
Control rod sticking	1
Multi pump failure	8, 9, 39, 40, 21, 22, 27, 28, 46, 62, 63
Multi valve failure	35, 36, 52, 53, 31, 32, 17, 66, 67
Pressurizer failure	2, 3, 4, 5
Pump failure and valve failure	10, 11, 69
Single feed water pump failure	12, 13, 15, 14
Single pump failure	6, 7, 37, 18, 48, 49, 19, 20, 25, 26, 68, 44, 45, 60, 61
Single valve failure	16, 24, 41, 42, 43, 55, 56, 57, 58, 59, 33, 34, 51, 29, 30, 64, 65

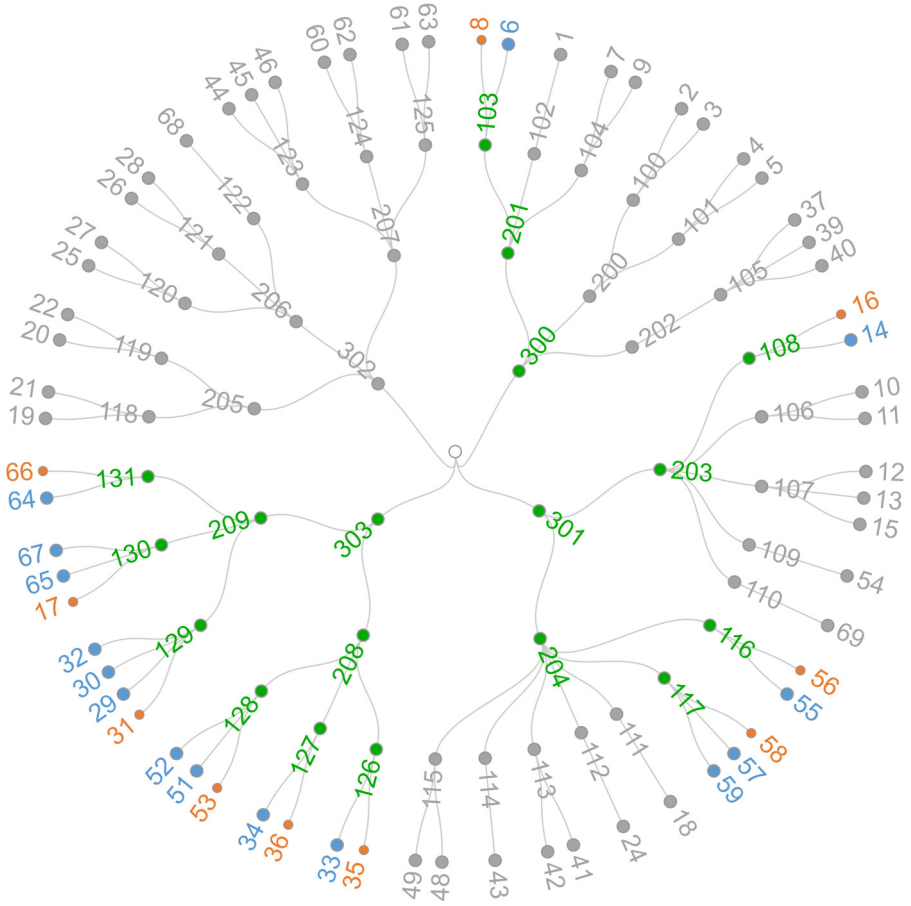


Fig. 3. Three superclass layers hierarchy of all fault types in FARON dataset. The orange small node represents the target class, the blue large node represents the source class selected for pre-training, the green node represents the superclass, and the gray node represents the abandoned source class. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Structure of the HFN on FARON dataset.

Layer number	HFN model
Input(signal)	$11 \times 11 \times 1$
l_1	Conv2d($9 \times 9 \times 64$)
l_2	Maxpool2d($8 \times 8 \times 64$)
l_3	Conv2d($4 \times 4 \times 4$)
l_4	Maxpool2d($3 \times 3 \times 4$)
Output(feature)	36×1

of convolution, as shown in Table 3. The 121-dimensional original sensor data are input in the form of 11×11 matrix, and a 36-dimensional feature vector is output after the feature extraction of HFN. During pre-training, the Adam optimization algorithm is applied with a base learning rate of 0.01. The mini-batch size, weight decay, step of learning rate decreases and attenuation factor are set to 32, 0.0005, 25, and 0.618, respectively.

3.1.4. Transfer learning for few shot fault recognition of target domain

After using the HFN to obtain the fault features from $S_{support}$, several widely used recognition methods are used to predict fault types, including k-nearest neighbor (KNN), logistic regression (LR), nearest centroid (NC), and support vector machine (SVM). All algorithms are implemented with sklearn toolkit, except NC. For KNN, the number of nearest neighbors is chosen as 1. For NC, each class is represented by the centroid, and the similarity between classes is measured by cosine distance. Other algorithm parameters are set by default. Finally, five HFN instances are trained by randomly splitting in S_{source}. Each model instance is applied to repeat 300 tests on $S_{support}$ and S_{test}. The mean weighted F1-score is used as the evaluation metric.

Table 4
Final test accuracy (as %) of all compared methods on the FARON dataset.

	n-shot	1	5	10	15	20	25
Baseline	LR	48.08	59.76	67.54	72.53	76.20	79.02
	KNN	47.73	59.90	66.00	70.05	73.14	75.48
	NC	48.06	54.90	57.35	58.74	59.73	60.51
	SVM	47.72	64.03	73.90	78.94	82.11	84.39
Transfer learning methods	FT [25]	52.21	73.34	80.04	82.63	84.33	85.35
	SJFT [26]	54.31	69.23	73.92	76.34	77.98	79.28
	LSFSL [23]	63.16	75.47	79.11	80.59	81.47	82.03
Ours	HFN_LR	68.42	**80.37**	**82.99**	**84.19**	**84.91**	**85.49**
	HFN_KNN	67.36	78.36	81.84	83.38	84.31	85.00
	HFN_NC	**68.86**	80.00	82.63	83.58	84.24	84.59
	HFN_SVM	66.97	77.69	78.85	78.92	79.24	80.16

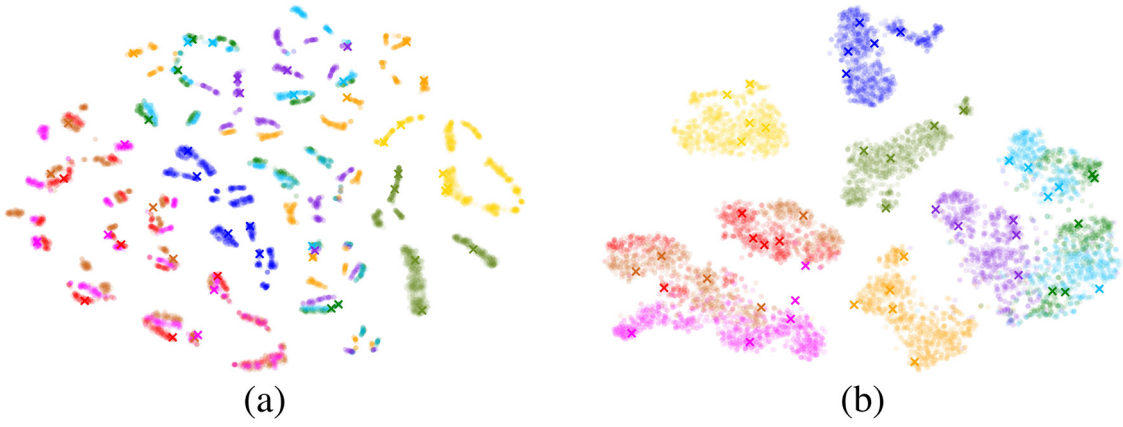


Fig. 4. Data visualization via t-SNE for (a) the original data, (b) transferable feature. There are 10 target domain classes and each color represent each class. Notation: 'x'— the training samples of each target class, 'o'— the test samples of each target class.

3.2. Comparison results

We compare our model with seven alternatives. Four of them are the general classification methods as the baseline, and the others are recent transfer learning methods to solve few-shot problems: FT fine-tuning pre-trained [25], SJFT selective joint fine-tuning [26], and LSFSL large scale few-shot learning [23]. Table 4 provides the comparative results on FARON dataset. It can be seen that our methods perform better than other methods. Taking 5-shot as an example, the classification accuracy of all baseline methods is less than 65%, the accuracy of transfer learning methods can reach more than 75%, and our best accuracy is about 80%. It is mainly caused by the following reasons.

In most cases, too few samples often cannot fully express the data distribution in the feature space, so the baseline method in the experiment is difficult to obtain a good classification interface. As we all know, CNN model has good feature extraction ability, but it needs rich samples to train available model. If there are only a few labeled samples, it is impossible to obtain an effective model directly.

With the help of the proposed hierarchy guided transfer learning method, we use the hierarchical relationship information of the categories to select the source domain categories which are similar to the target domain categories. These categories have abundant labeled samples, which can help us to obtain the available feature extraction network. In Fig. 4, the data distribution of original feature data and portability feature are visualized by t-SNE. It can be seen that the transferable features extracted from the network are more clustered and the separability between classes is more significant. Data distribution can be better represented even though there are only a few samples.

The FT model first carries on the pre-training with the help of numerous source domain samples, and then uses the target domain support samples to fine-tune the parameters. Many studies have proved that this is a simple and effective method to solve the shortage of samples. However, for tasks with very little training data, such as 1-shot and 5-shot, overfitting occurs very quickly during fine-tuning. In addition, the transfer capacity of different convolution layers is different, so it is a problem to select which layers.

Based on the idea of data augmentation, SJFT model performs a target learning task with insufficient training data and another source learning task with rich training data at the same time. The source learning task uses a subset of training data from the source task whose low-level features are similar to those from the target task, and jointly fine-tune shared convolution layers of the two tasks. However, similar samples may come from multiple categories of the source domain. The more similar samples are selected, the greater the feature deviation from the target class. Joint fine-tuning may further drift the feature expression of the target domain to the feature space of the source domain, thus reducing the performance on the target task. For this reason, the performance of SJFT model on FARON dataset is worse than that of FT.

The LSFSL model uses the hierarchical information between classes based on semantic relations, but it uses the source domain samples of all classes for pre-training, ignoring the impact of negative transfer on the target task. It also ignores the differences of each superclass level transfer capabilities. Therefore, the results of LSFSL model are the best among the transfer learning alternatives, but worse than our methods, indicating that hierarchical information transfer is effective, but will be affected by negative transfer.

Our method makes full use of a variety of prior knowledge to construct a more effective class hierarchy. When obtaining the feature expression of the target class in the pre-training stage, we can not only effectively alleviate the impact of negative transfer, but also take into account the transfer ability of different granular-

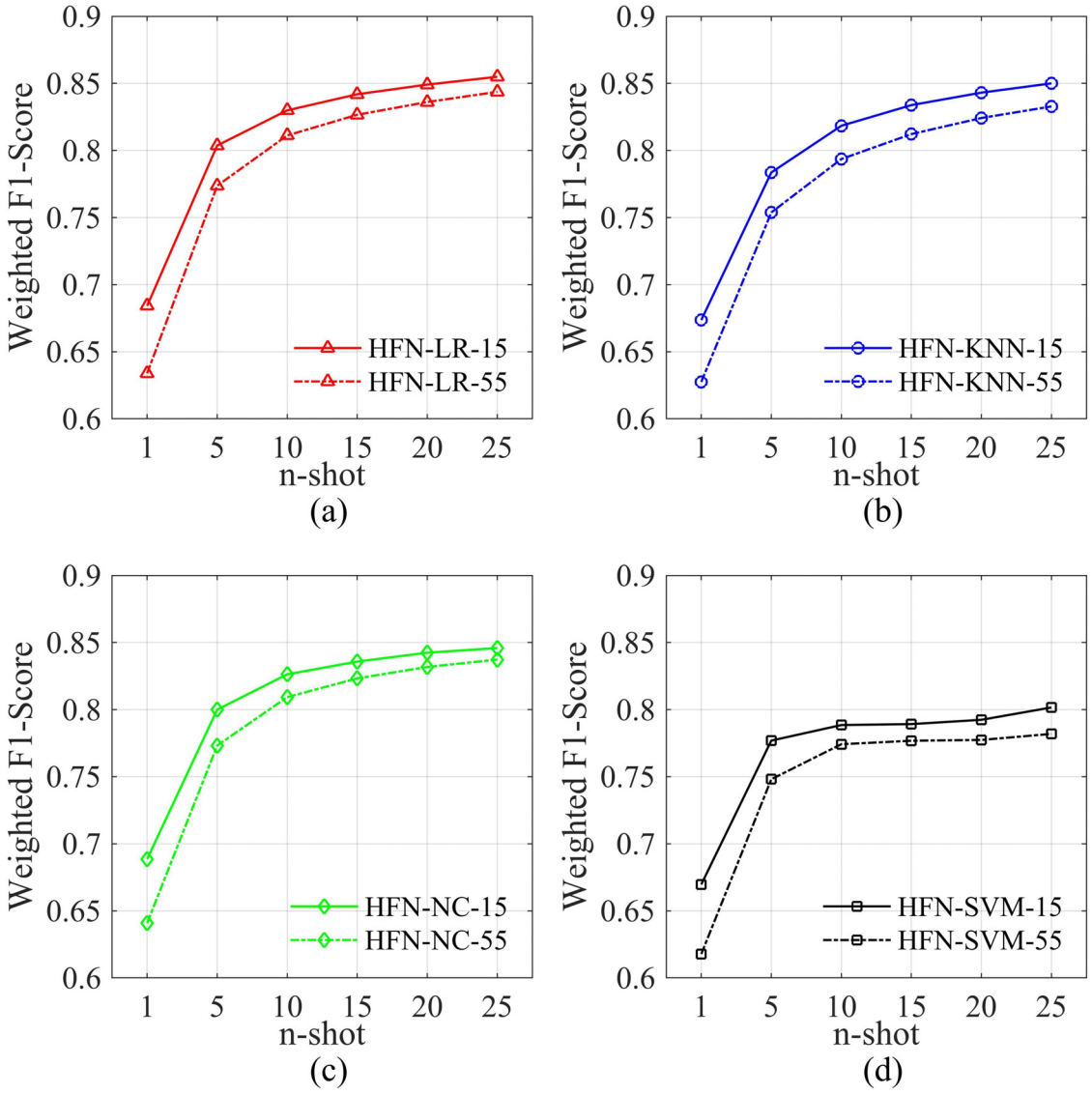


Fig. 5. Comparative results obtained by four classification algorithms, when pre-training using selected 15 class samples and all 55 class samples in source domain. Results were obtained using a HFN with $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) = (0.6, 0.25, 0.1, 0.05)$ based on 3 superclass layers hierarchy.

ity levels, so it obtains better performance. In the final reasoning stage, the use of general classification algorithm for fault recognition can simplify model training and parameter adjustment, which is beneficial to practical application.

Fig. 5 show the performance results when the hierarchical selection policy for source domain class is used or not based on the hierarchy tree. It can be seen that under the condition of few samples, using the hierarchical selection strategy can achieve better performance than not using it. In the test of different support samples, the performance improvement rate is about 1% $\sim$ 8%. This is mainly caused by the following reasons.

Because the source domain category is different from the target domain category, only part of the category information in the source domain has guiding significance for the target domain classification task. In transfer learning, there is a negative transfer phenomenon. When all the source domain class data are used to train the feature extraction model, some source domain categories with poor correlation with the target domain will lead to the reduction of the classification ability of the extracted features on the target domain. On the other hand, the ability of extracting effective features decreases with the increase of the number of categories to be processed. The more the number of source domain categories, the more difficult the training of feature extraction model is. For the target domain classification task, the ability of extracting effective features will also decline. Therefore, after the selection of source domain categories, the number of categories faced by the feature extraction model can be reduced, the difficulty of feature extraction can be reduced, and the extraction ability of effective features can be improved accordingly.

The experimental results show that our method can extract the transferable features of the target samples even without the target samples. This means that learning feature embedding can be generalized to unknown classes. It can be expected that various information about category relevance can be encoded into the class hierarchy, which makes it possible to learn the transferable features of the target class sample. In short, the information contained in the hierarchical relationship is transferable, so the learned features are also transferable.

3.3. Hyperparameter selection

There are two important hyperparameters in the experiment, one is the number of superclass layers, the other is λ_i of each layer for loss function. First, we construct three class hierarchies with

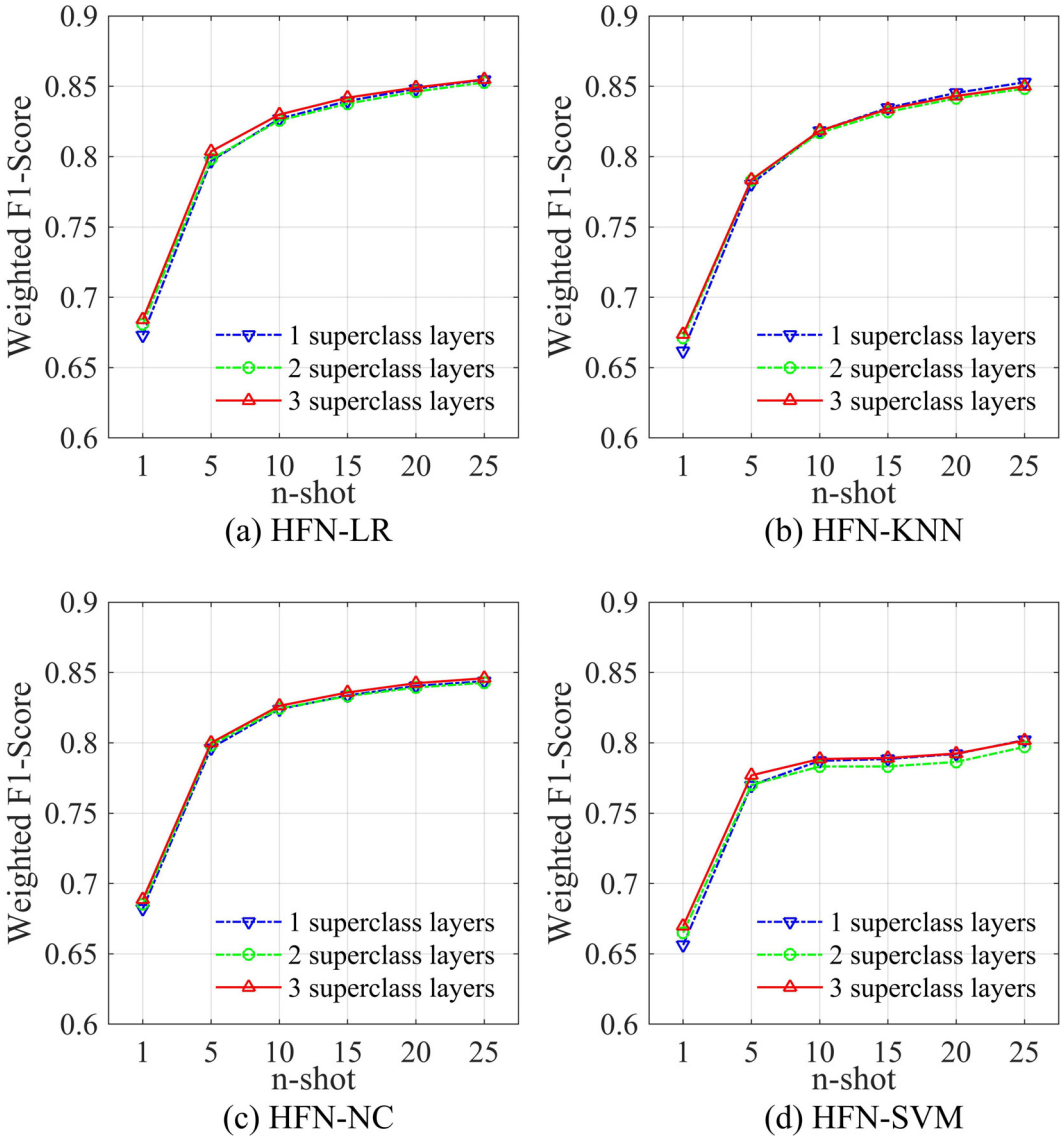


Fig. 6. Comparative results obtained by our model using the hierarchies with different numbers of superclass layers on the FARON dataset.

Table 5
The number of superclasses per layer.

No.	l_0	l_1	l_2
1	32	-	-
2	32	10	-
3	32	10	4

Table 6
Analysis Results with different λ_i on FARON dataset.

No.	λ_0	λ_1	λ_2	λ_3
1	0.1	0.2	0.3	0.4
2	0.25	0.25	0.25	0.25
3	0.4	0.3	0.2	0.1
4	0.6	0.25	0.1	0.05

different structures for comparison, and then use them to train our feature extraction model. Table 5 gives details of the three class hierarchies. Fig. 6 shows the comparison results of the above three class hierarchies in our work. The mean weighted F1-score of the target class is used as the evaluation metric. It can be observed that a class hierarchy with 3 superclass layers produces the best results, but the gap is not very obvious.

After determining the optimal number of superclass layers, the loss weight λ_i of each level is selected by experiment. We gradually reduce the weight of low-level loss and increase the weight of high-level loss accordingly. Table 6 gives details of the value of λ_i. Fig. 7 shows the comparison results of different weights when we use three superclass levels. It can be seen that the best performance occurs when $\lambda_i (i = 0, 1, 2, 3)$ is taken as (0.6, 0.25, 0.1, 0.05). From the current experimental results, the higher the weight of the lower level is, the better the performance is. The reason is that the lower the category level is, the more categories there are.

Especially for the source class in the leaf layer, the features it contains are most similar to the target class, so a higher weight should be set. During the process of training optimization, the higher the level of superclasses, the less number of superclasses, and the easier it is for the subtask model of the corresponding layer to converge. However, the higher level contains more common features between classes, and lacks the distinguishing features, so λ_i of the higher-level should be set a smaller value.

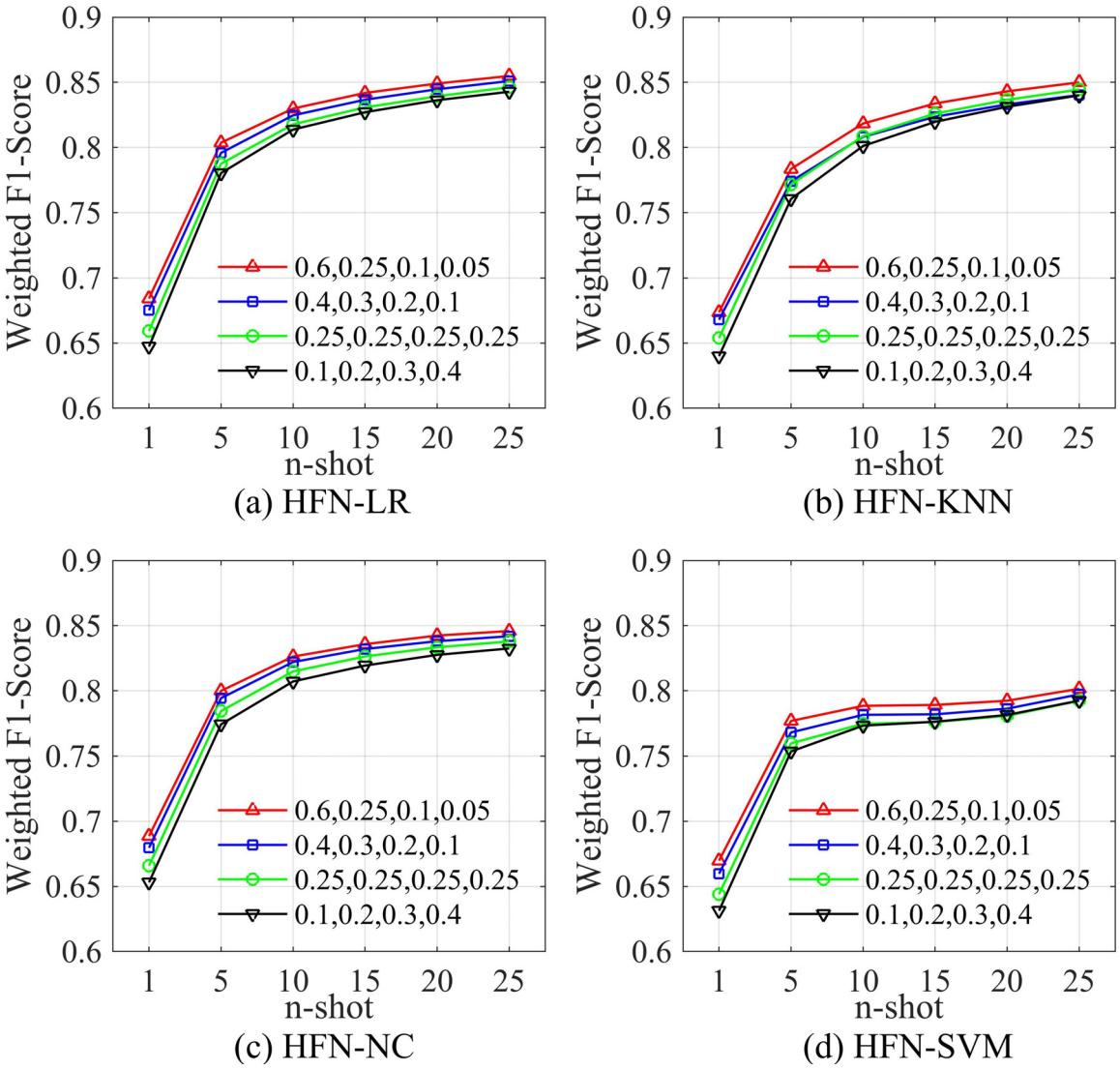


Fig. 7. Comparative results obtained by our model using different λ_i on the FARON dataset. The hierarchy has three superclass layers.

3.4. Sample size sensitivity analysis

Fig. 8 shows the performance trend of our method and other alternatives as the number of samples increases gradually on the FARON dataset. It can be seen from Fig. 8 that the fewer samples, the more improvement our method has on the target domain classification performance. Even when $n = 1$, our method can still achieve a weighted F1-score of more than 67%, while all baseline methods are below 50%. The reason is that when the number of training samples in the target domain is too small to fully express the data distribution, the transferable feature information obtained from the source class with abundant samples can provide effective help. With the increase of the number of samples, the performance of all methods is gradually improved. When $n \geq 30$, the performance of SVM is better than other methods. The reason may be that when the number of training samples in the target domain is enough to express the data distribution, the effect of the transferable information obtained from the source domain class is relatively weakened. As a recognized and effective algorithm, SVM can generate a good classification hyperplane with sufficient information provided by the training sample itself. The above analysis shows that our method is effective under the condition of few samples. It can get extra information from the source domain data which is helpful to the target domain classification.

4. Case study 2

4.1. Experimental setup

4.1.1. Data description

We also use TE dataset for verification experiments. TE process is a simulation control process established by Tennessee Eastman Chemical Company based on the actual industrial process. It is a common chemical benchmark for fault detection and process control research. TE process is mainly composed of reactor, condenser, vapor-liquid separator, compressor, and stripper. Four kinds of reactants A, C, D and E are fed into the reactor to form products G and H by exothermic reaction. In TE process, there were 52 variables measured, including 22 continuous process measurements (XMEAS 1∼22), 19 composition measurements (XMEAS 23∼41), and 11 manipulated variables (XMV 1∼11). Twenty-one fault states are simulated in TE dataset. The sampling interval of each sample is 3 min, and random noise is added to each simulation run.

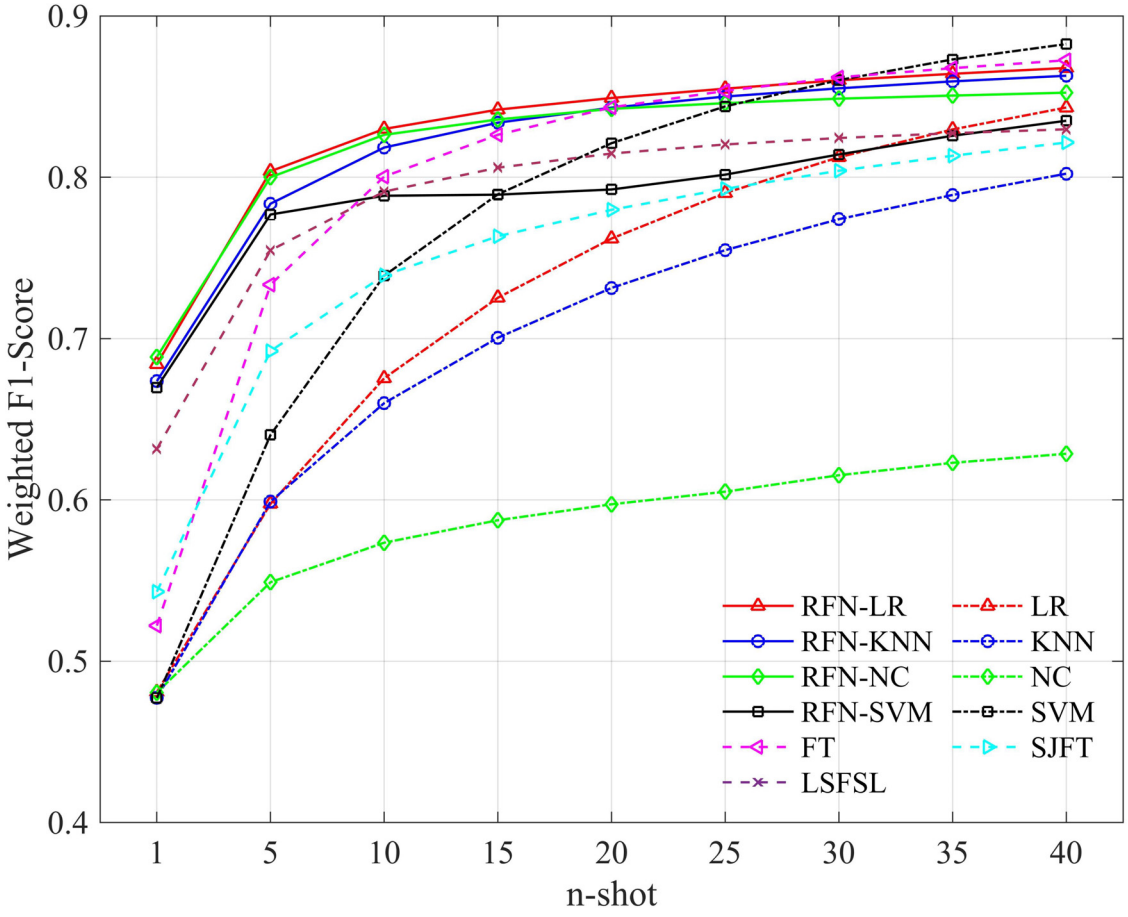


Fig. 8. Comparative the performance trend obtained by our method and alternatives as the number of samples increases gradually on the FARON dataset. The hierarchy has 3 superclass layers and $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$=(0.6, 0.25, 0.1, 0.05).

For each fault state, training and test datasets are collected respectively. In the training set, 480 samples were recorded for each fault. In the test set, 960 samples are collected for each fault, of which the first 160 are normal state samples and the last 800 are fault state samples.

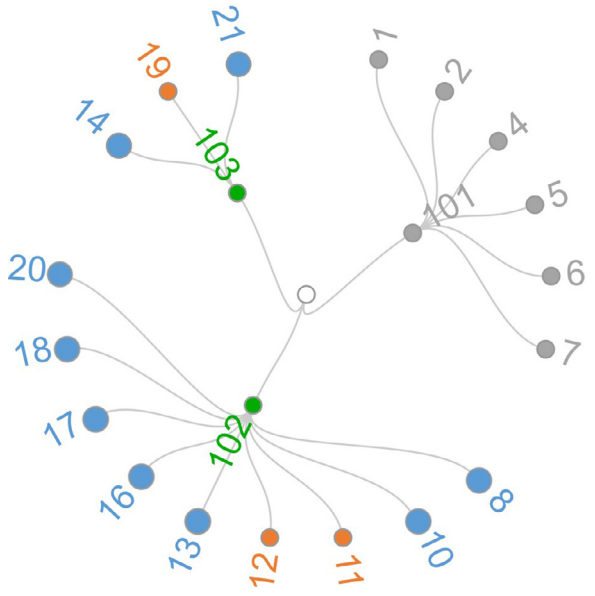


Fig. 9. The hierarchical fault category structure of TE dataset.

Table 7
Fault types on TE dataset.

Type	Fault
Step	1, 2, 3, 4, 5, 6, 7
Random fluctuation	8, 9, 10, 11, 12, 16, 17, 18, 20
Slow drift	13
Sticking	14, 15, 19
Constant position	21

In our experiment, faults 3, 9 and 15 are removed and the remaining 18 faults are divided into two parts. Faults 11, 12 and 19 constitute the target domain, and the remaining constitute the source domain. Each sample is organized into a 33×33 matrix. There are 33 variables in each row, including 22 process measurements (XMEAS 1∼22) and 11 manipulated variables (XMV 1∼11). The size of sliding window is 33 and the sliding step is 1. Finally, for each fault, the size of training set is $33 \times 33 \times 448$, and that of test set is $33 \times 33 \times 768$.

4.1.2. Hierarchical fault category structure

Table 7 lists all TE process fault types. According to domain knowledge, random fluctuation and slow drift are similar, while sticking and constant position faults are similar. Therefore, only using above knowledge, we construct a fault class hierarchy containing 2 superclasses from 18 types of faults in the experimental dataset, as shown in Fig. 9.

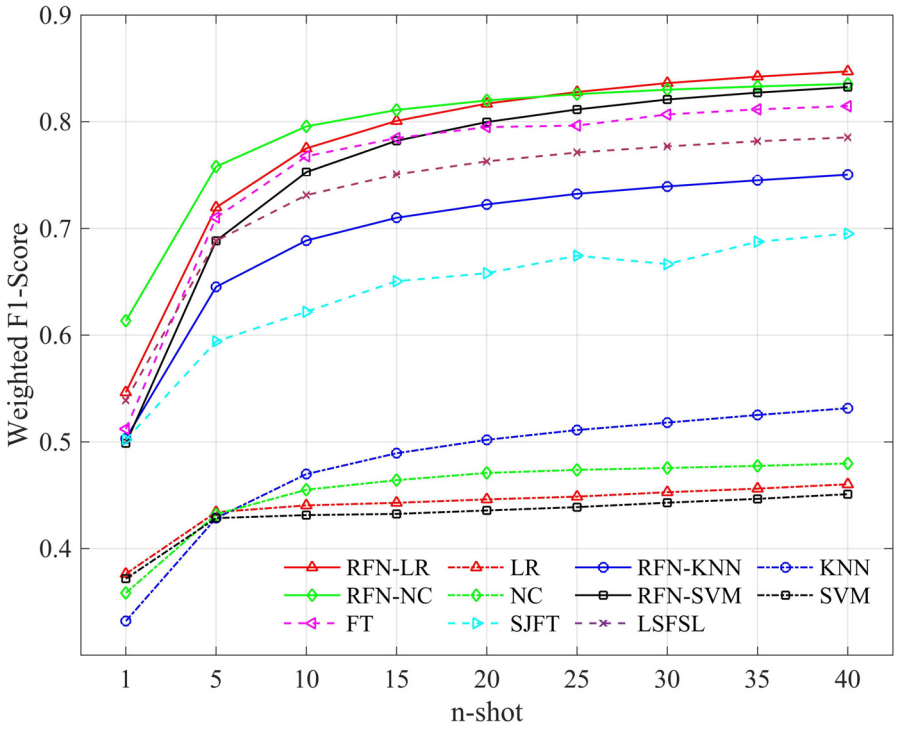


Fig. 10. Comparison Results of HFN with $(\lambda_0, \lambda_1) = (0.85, 0.15)$ was performed by using 9 categories of source domain samples.

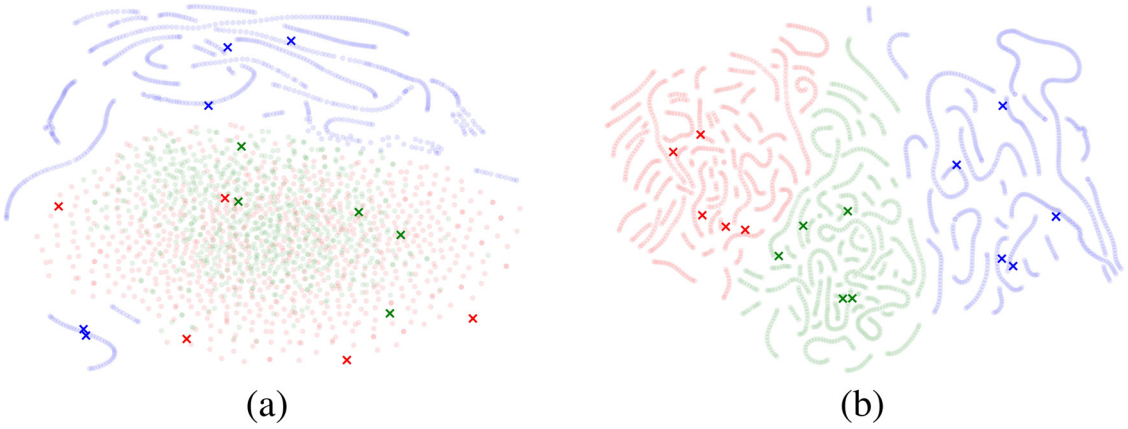


Fig. 11. Data visualization via t-SNE for (a) the original data, (b) transferable feature. There are 3 target domain classes and each color represent each class. Notation: 'x'— the training samples of each target class, 'o'— the test samples of each target class.

4.1.3. Pre-training details of source domain

We select 9 types of fault (blue node) samples in superclass 102 and 103 (green node) which contain fault 11, 12 and 19 (orange nodes) as the source domain data class samples to perform pre-training on HFN, and discard the source domain class (gray node) samples under superclass 101. The feature extractor is mainly composed of three layers of convolution, as shown in Table 8. Each input sample data is organized into a 33×33 matrix. After HFN, a 484-dimensional feature vector can be output. During pre-training, the Adam optimization algorithm is applied with a base learning rate of 0.01. The mini-batch size, weight decay, step of learning rate decreases, attenuation factor and max epoch are set to 128, 0.0005, 25, 0.618 and 500, respectively.

4.1.4. Transfer learning for few shot fault recognition of target domain

The experimental method of few-shot fault recognition in the target domain of TE dataset is basically the same as the previous Section 3.1.4 of case study 1. The difference is that n samples ($n \leq 40$) are randomly selected from the training set of each fault as the support set, and the classification performance evaluation is conducted using the test set samples of each fault. Finally, five HFN

Table 8
Structure of the HFN on TE dataset.

Layer number	Parameters
Input(signal)	$33 \times 33 \times 1$
l_1	Conv2d($31 \times 31 \times 64$)
l_2	Maxpool2d($30 \times 30 \times 64$)
l_3	Conv2d($28 \times 28 \times 32$)
l_4	Maxpool2d($27 \times 27 \times 32$)
l_5	Conv2d($23 \times 23 \times 1$)
l_6	Maxpool2d($22 \times 22 \times 1$)
Output(feature)	484×1

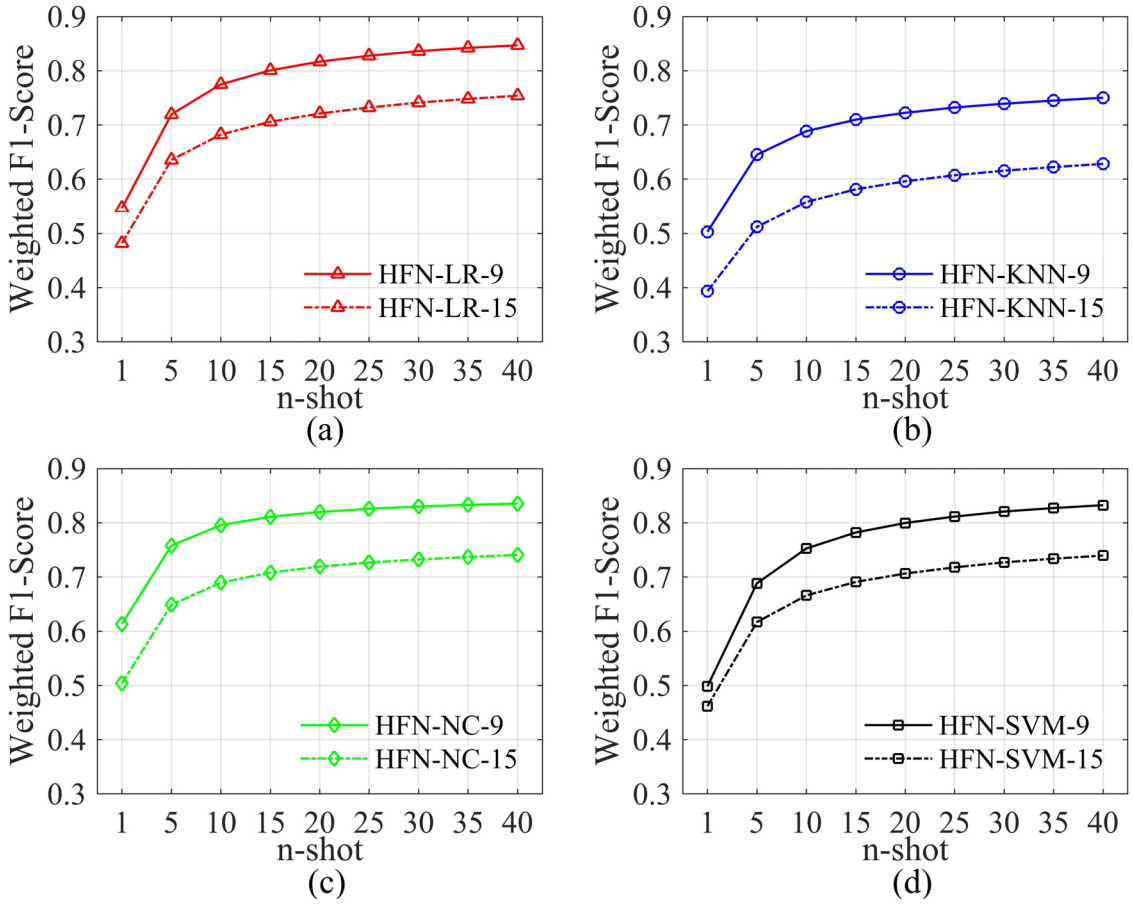


Fig. 12. Comparative results obtained by four classification algorithms, when pre-training using the selected 9 class samples and all 15 class samples in source domain.

Table 9
Final test accuracy (as %) of all compared methods on the TE dataset.

	n-shot	1	5	10	15	20	25	30	35	40
Baseline	KNN	33.21	42.85	46.97	48.93	50.19	51.10	51.80	52.51	53.15
	LR	37.63	43.41	44.04	44.29	44.60	44.86	45.28	45.61	46.02
	NC	35.84	43.21	45.51	46.41	47.08	47.37	47.55	47.74	47.96
	SVM	37.17	42.85	43.13	43.25	43.57	43.88	44.29	44.65	45.09
Transfer learning method	FT [25]	51.22	71.02	76.75	78.47	79.49	79.63	80.68	81.16	81.46
	SJFT [26]	50.19	59.41	62.19	65.05	65.80	67.44	66.65	68.74	69.52
	LSFSL [23]	53.88	68.84	73.12	75.06	76.28	77.11	77.68	78.17	78.53
Ours	HFN_KNN	50.31	64.52	68.86	70.99	72.25	73.24	73.93	74.51	75.03
	HFN_LR	54.65	71.96	77.50	80.07	81.69	**82.78**	**83.62**	**84.22**	**84.71**
	HFN_NC	**61.34**	**75.79**	**79.56**	**81.10**	**81.99**	82.59	83.00	83.30	83.54
	HFN_SVM	49.86	68.84	75.27	78.21	79.97	81.14	82.08	82.72	83.24

instances are trained by randomly seed. Each model instance is applied to repeat 300 tests on $S_{support}$ and S_{test}. The mean weighted F1-score is used as the evaluation metric.

4.2. Comparative results

Fig. 10 and Table 9 show the performance comparison results of the seven alternatives and our methods on TE dataset. It can be seen that our method can greatly improve the performance of baseline method and better than other transfer learning methods on TE dataset.

In Fig. 11, t-SNE visualizes the 2D data distribution of original feature and transferable features. It can be seen that the class separability of the transferable features extracted by HFN is obviously better than that of the original features.

Fig. 12 shows the performance comparison results of four classification methods for few-shot classification when using 9 selected source domain class samples and all 15 source domain class samples, respectively. It can be seen that in the experiment of TE dataset, the performance can be improved by about 8%∼28% when the source domain selection strategy is used.

4.3. Hyperparameter selection

In the experiment of TE data set, the number of super classes is fixed to 1. Fig. 13 shows the performance comparison results of different weights. Table 10 gives the details of the loss weight λ_i of each level. It can be seen that the best performance occurs when $\lambda_i (i = 0, 1)$ is taken as (0.85, 0.15).

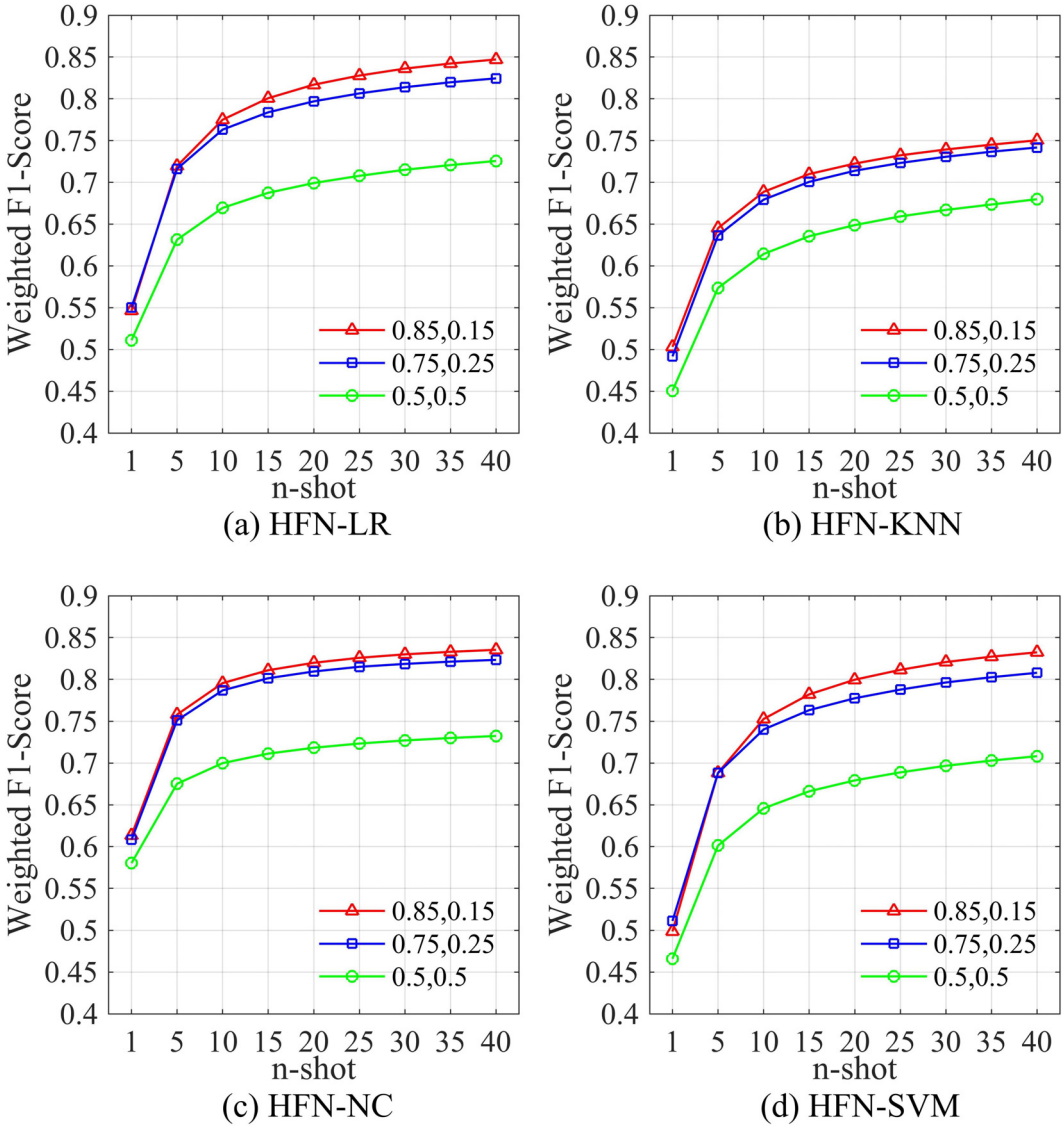


Fig. 13. Comparative results obtained by our model using different λ_i on the TE dataset. The hierarchy has only one superclass layers.

Table 10
Analysis Results with different λ_i on TE dataset.

No.	λ_0	λ_1
1	0.5	0.5
2	0.75	0.25
3	0.85	0.15

5. Conclusion

To solve the few-shot fault recognition problem in complex industrial systems, a novel hierarchy guided transfer learning framework, HGTL, is proposed in this article. Considering that transfer learning can be an effective tool for few-shot recognition, but not all faults in a complex system are suitable to transfer, a hierarchical category structure is firstly constructed to select the similar faults as source domain classes. Then, the model is pre-trained with these source class samples, which make it possible to extract the fault features of target fault within few samples. Finally, the experimental and comparative results show that our method has better performance in the case of few samples.

However, the proposed framework still has many problems worthy of in-depth study. For example, in the stage of constructing class hierarchy, artificial experience is used to weight and fuse multi-source prior knowledge. Unreasonable weight makes it impossible to obtain the optimal hierarchy. The optimal weight on the target task can be learned automatically through the data-driven method. In addition, this framework adopts a phased processing flow and has not yet achieved global optimization, so it can be improved into an end-to-end learning framework in the future.

Few-shot fault recognition is always a practical and difficult problem. Though the proposed method performs well in these two datasets, its generalization capability still should be further explored. In the future, we will apply our approach to more few-shot fault recognition tasks and improve it accordingly.

Declaration of Competing Interest

None.

Acknowledgements

This work is supported by National Science and Technology Major Project 2017-I-0007-0008, National Natural Science Foundation of China under Grants 61925602 and 61732011, and the key R&D

Program of Science and Technology Foundation of Hebei Province 19210310D.

References

[1] B. Cai, H. Liu, M. Xie, A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks, Mech. Syst. Signal Process. 80 (2016) 31–44.

[2] T. Han, C. Liu, L. Wu, S. Sarkar, D. Jiang, An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems, Mech. Syst. Signal Process. 117 (2019) 170–187.

[3] L. Wang, Z. Liu, H. Cao, X. Zhang, Subband averaging kurtogram with dual-tree complex wavelet packet transform for rotating machinery fault diagnosis, Mech. Syst. Signal Process. 142 (2020) 1–21.

[4] H. Wang, M.j. Peng, J.W. Hines, G.y. Zheng, Y.k. Liu, B.R. Upadhyaya, A hybrid fault diagnosis methodology with support vector machine and improved particle swarm optimization for nuclear power plants, ISA Trans. 95 (2019) 358–371, doi:10.1016/j.isatra.2019.05.016.

[5] F. De Vita, D. Bruneo, S.K. Das, On the use of a full stack hardware/software infrastructure for sensor data fusion and fault prediction in industry 4.0, Pattern Recognit. Lett. 138 (2020) 30–37, doi:10.1016/j.patrec.2020.06.028.

[6] R. Liu, B. Yang, E. Zio, X. Chen, Artificial intelligence for fault diagnosis of rotating machinery: a review, Mech. Syst. Signal Process. 108 (2018) 33–47.

[7] L. Yao, Z. Ge, Industrial big data modeling and monitoring framework for plant-wide processes, IEEE Trans. Ind. Inf. (2020) 1, doi:10.1109/TII.2020.3010562.

[8] G. Bhatt, P. Jha, B. Raman, Representation learning using step-based deep multi-modal autoencoders, Pattern Recognit. 95 (2019) 12–23, doi:10.1016/j.patcog.2019.05.032.

[9] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Trans. Smart Grid 10 (1) (2017) 841–851.

[10] R. Liu, F. Wang, B. Yang, S.J. Qin, Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions, IEEE Trans. Ind. Inf. 16 (6) (2020) 3797–3806.

[11] C. Li, S. Zhang, Y. Qin, E. Estupinan, A systematic review of deep transfer learning for machinery fault diagnosis, Neurocomputing 407 (2020) 121–135, doi:10.1016/j.neucom.2020.04.045.

[12] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2021) 43–76, doi:10.1109/JPROC.2020.3004555.

[13] B. Yang, Y. Lei, F. Jia, S. Xing, An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings, Mech. Syst. Signal Process. 122 (2019) 692–706.

[14] Z. Liu, B. Lu, H. Wei, L. Chen, X. Li, M. Rätsch, Deep adversarial domain adaptation model for bearing fault diagnosis, IEEE Trans. Syst. Man Cybern. (99) (2019) 1–10.

[15] Y. Wang, D. Wu, X. Yuan, LDA-based deep transfer learning for fault diagnosis in industrial chemical processes, Comput. Chem. Eng. (2020) 106964.

[16] J. Li, R. Huang, G. He, S. Wang, G. Li, W. Li, A deep adversarial transfer learning network for machinery emerging fault detection, IEEE Sens. J. 20 (15) (2020) 8413–8422, doi:10.1109/JSEN.2020.2975286.

[17] W. Qian, S. Li, X. Jiang, Deep transfer network for rotating machine fault analysis, Pattern Recognit. 96 (2019) 106993, doi:10.1016/j.patcog.2019.106993.

[18] M.J. Afridi, A. Ross, E.M. Shapiro, On automated source selection for transfer learning in convolutional neural networks, Pattern Recognit. 73 (2018) 65–75, doi:10.1016/j.patcog.2017.07.019.

[19] L. Dong, S. LIU, H. ZHANG, A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples, Pattern Recognit. 64 (2017) 374–385, doi:10.1016/j.patcog.2016.11.026.

[20] J. Wu, Z. Zhao, C. Sun, R. Yan, X. Chen, Few-shot transfer learning for intelligent fault diagnosis of machine, Measurement 166 (2020) 108202, doi:10.1016/j.measurement.2020.108202.

[21] Y. Qu, L. Lin, F. Shen, C. Lu, Y. Wu, Y. Xie, D. Tao, Joint hierarchical category structure learning and large-scale image classification, IEEE Trans. Image Process. 26 (9) (2017) 4331–4346, doi:10.1109/TIP.2016.2615423.

[22] Y.H. Tsai, R. Salakhutdinov, Improving one-shot learning through fusing side information, CoRR (2018) abs/1710.08347.

[23] A. Li, T. Luo, Z. Lu, T. Xiang, L. Wang, Large-scale few-shot learning: knowledge transfer with class hierarchy, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7212–7220.

[24] G. Xie, J. Yang, Y. Yang, An improved sparse autoencoder and multilevel denoising strategy for diagnosing early multiple intermittent faults, IEEE Trans. Syst. Man Cybern. (2020) 1–12, doi:10.1109/TSMC.2020.3005433.

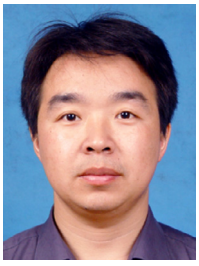
[25] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, doi:10.1109/CVPR.2014.81.

[26] W. Ge, Y. Yu, Borrowing treasures from the wealthy: deep transfer learning through selective joint fine-tuning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 10–19, doi:10.1109/CVPR.2017.9.

Hao Chen received the B.S. degree in Computer software and M.S. degree in Mathematics from Hebei University, in 1999 and 2005 respectively. He is an associate professor in the College of Mathematics and Information Science, Hebei University. He currently is a Ph.D. student in school of Computer Science and Technology, Tianjin University. His re- search interests include machine learning, pattern recognition, and data mining for hierarchical classification.

Ruonan Liu (M'19) received the B.S., M.S. and PhD 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 in 2019. She currently is an associate professor in the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include machine learning, intelligent manufacturing and computer vision.

Zongxia Xie received her B.S. from Dalian Maritime University in 2003, and M.S. and Ph.D. from Harbin Institute of Technology in 2005, and 2010, respectively. She has worked as postdoctoral researcher at Shenzhen Graduate School, Harbin Institute of Technology, from Dec. 2010 to Jan. 2013. Now she is an associate professor at College of Intelligence and Computing in Tianjin University. Her major interests include ma- chine learning, pattern recognition, especially. She has published more than 20 conference and journal papers on related topics.



Qinghua Hu (M'13) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He was a Post-Doctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, from 2009 to 2011. He is currently the Dean of the School of Artificial Intelligence, the Vice Chairman of the Tianjin Branch of China Computer Federation, the Vice Director of the SIG Granular Computing and Knowledge Discovery, and the Chinese Association of Artificial Intelligence. He is currently supported by the Key Program, National Natural Science Foundation of China. He has published over 200 peer-reviewed papers. His current research is focused on uncertainty modeling in big data, machine learning with multi-modality data, intelligent unmanned systems. He is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, Acta Automatica Sinica, and Energies.

Jianhua Dai received the B.Sc., M.Eng., and Ph.D. degrees in computer science and technology from Wuhan University, Wuhan, China, in 1998, 2000, and 2003, respectively. He is currently the Director of the Hunan Provincial Key Laboratory of Intelligent Computing and Language In- formation Processing and the Dean of the College of Information Science and Engineering, Hunan Normal University, Changsha, China. His current research interests include artificial intelligence, machine learning, and intelligent information processing.

Junhai Zhai (Member, IEEE) received the B.S. degree in mathematics and the M.S. degree in computing mathematics from Lanzhou University, Lanzhou, China, in 1988 and 2000, respectively, and the Ph.D. degree in optical engineering from Hebei University, Baoding, China, in 2010. He is currently a Professor with the College of Mathematics and Information Science, Hebei University. His main research interests include machine learning, deep learning, and big data processing.