Distributed Hierarchical Fault Diagnosis Based on Sparse Auto-Encoder and Random Forest

Tong Li[1], Chunhe Song[2,3(✉)], Yang Liu[1], Zhongfeng Wang[2,3], Shimao Yu[2,3], and Shanting Su[4]

[1] Liaoning Electric Power Research Institute, State Grid Liaoning Electric Power Co., Ltd., Shenyang 110000, People's Republic of China
[2] Key Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, People's Republic of China
songchunhe@sia.cn
[3] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[4] Nanjing University of Aeronautics and Astronautics, Nanjing, China

Abstract. For the diagnosis of large-scale local devices, the traditional centralized fault diagnosis systems are becoming incompetent to meet the requirement of real-time monitoring. This paper proposes the Distributed hierarchical Fault Diagnosis System (DFDS). Specifically, DFDS implements fault monitoring by an improved Sparse Auto-Encoder (SAE) on the monitor layer, classifies faults and identifies unknown faults by an improved random forest on the classification layer, learns new knowledge and updates the system on the decision layer. We apply DFDS in the laboratory data of Case Western Reserve University to verify the efficiency of the proposed system. The experimental results show that our method can accurately detect the fault and accurately identify the fault type.

Keywords: Sparse auto-encoder · Distributed fault diagnosis · Fault classification · Random forest

1 Introduction

Condition monitoring is essential for safe and reliable working operation of electric power system such as transformer, GIS and High voltage circuit breaker. In recent years, artificial intelligence (AI) technology has developed rapidly, and many AI based methods have been developed to solve equipment failure problems such as neural networks [1, 2], fault trees [3, 4], fuzzy theory [5, 6] and deep learning [7–9].

The traditional fault diagnosis method generally collects raw device information of multiple local sensors and uploads them to the terminal device. The terminal device extracts the original device signal characteristics and performs intelligent diagnosis. The diagnostic complexity of this process is usually high and takes a lot of time. In addition, each device has different faults and different fault handling methods. In this case, the fault diagnosis system is required to identify the fault on each device to

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 244–255, 2019.
https://doi.org/10.1007/978-3-030-32388-2_21

achieve the purpose of dynamic real-time monitoring. After the fault is identified, it can be processed in time. The traditional centralized fault diagnosis system cannot meet the above challenges.

This paper proposes a distributed artificial intelligence fault diagnosis system that is superior to the traditional centralized fault diagnosis system in large-scale equipment fault diagnosis. The proposed multi-level hierarchical fault diagnosis model is to monitor the original information in real time through the Sparse Auto-Encoder [16] (SAE) on the local monitoring layer (i.e., each device sensor), and report the fault signal for further processing. The improved random forest model is used in the fault classification layer to classify the detected faults and identify the unknown faults, which are uploaded to the decision-making layer for further processing. In summary, the contributions of this article are as follows:

(1) A novel distributed fault diagnosis framework and its implementation are proposed, which map multiple stages of device diagnosis to a distributed multi-level hierarchy.
(2) The fault monitoring model FM-SAE is embedded on the local device for large-scale real-time monitoring.
(3) Accurate classification of known faults and identification of unknown faults is achieved through an improved random forest model.

The rest of this paper is organized as follows. Section 2 outlines the architecture of the proposed distributed hierarchical fault diagnosis system. Section 3 introduces the implementation of the fault monitoring function of the model. Section 4 introduces the implementation of fault classification and update functions. Section 5 shows experimental results and analysis. Section 6 concludes this paper.

2 System Outline

In this section, we outline the architecture of the proposed Distributed hierarchical Fault Diagnosis System (DFDS) and describe how the system completes the process of device fault diagnosis through inter-level cooperation.

Equipment fault diagnosis is generally divided into a series of processes such as feature extraction, state detection, and fault classification. Due to the wide distribution of equipment, in order to provide timely fault monitoring for these devices, the primary layer of DFDS should be distributed on each device to reduce the complexity of the entire diagnostic process by making decisions directly on the original signal. Considering the diversity of equipment failures, if necessary, the secondary diagnostic layer subsystems can cooperate with each other and jointly diagnose, and the detected unknown faults should be reported to the superior diagnostic layer in time. Since the faults of each device are different, and the identified unknown faults are different, the third-level diagnostic layer should process the unknown faults in time to update the diagnostic knowledge of the secondary diagnostic layer subsystem, thereby improving the overall diagnostic level.

Based on the above starting point, the DFDS diagnostic system is divided into three layers: local monitoring layer, fault classification layer and decision-making layer.

Each of the three layers processes the respective tasks, and simultaneously communicates with each other to achieve the purpose of fault diagnosis. As shown in Fig. 1:

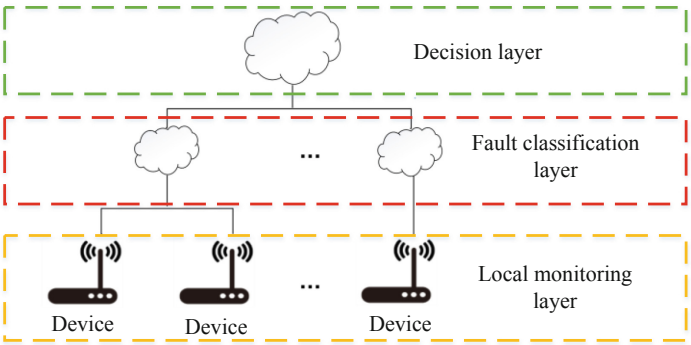

Fig. 1. The framework of DFDS.

In DFDS, The local monitoring layer is mainly responsible for detecting faults and reporting them to the fault classification layer in time. Directly process the original signal through the improved Fault Diagnosis Sparse Auto-Encoder (FM-SAE) model to determine if the device is faulty. Real-time monitoring of local devices is implemented without sending any information to the higher level diagnostic layer unless a fault signal is detected. For the detected fault signal, send it to the fault classification layer.

The fault classification layer is responsible for feature extraction of the received signals, classifying the faults and identifying unknown faults by improving the random forest. The fault classification layer may separately manage the local devices by multiple terminal devices, or may simultaneously manage multiple local devices by only one terminal device. They form multiple task nodes on the same level, they deal with problems independently of each other and are related to each other on the model, reducing the complexity of the diagnostic task.

The decision layer is responsible for the self-learning and updating of the entire system knowledge. By processing the unknown fault signal sent from the fault classification layer, learn new knowledge and update the system in real time.

DFDS completes the entire diagnostic process by implementing fault monitoring locally, classifying and updating at the terminal. Its various levels process their own tasks and coordinate with each other to achieve maximum fault diagnosis accuracy.

3 Fault Monitoring

This section describes the first phase of the model, the fault monitoring function, which is implemented by the local monitoring layer based on the improved sparse auto-encoder FM-SAE.

3.1 Theory of the Sparse Auto-Encoder Algorithm

The automatic encoder is a classic neural network whose purpose is to obtain the reduced-dimensional expression of the data $H = \{h_1, h_2, \ldots, h_m\}$ through training based on the input data set $X = \{x_1, x_2, \ldots, x_m\}$. This dimension-reduced feature is processed so that it can express the input better, and it has good performance in tasks such as classification monitoring [10–12]. The network structure of a single hidden layer AE is shown in Fig. 2, where the self-encoder is divided into two parts, the encoder and the decoder.

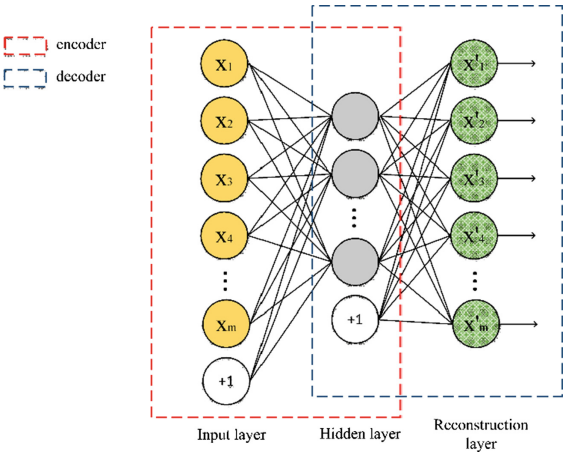

Fig. 2. A single hidden layer AE network structure.

Assume $a_i^{(l+1)}$ represents the activation value of the *i-th* neuron in the $(l + 1)$-*th* layer of the self-encoding neural network, then $a_i^{l+1} = f\left(z_i^{(l+1)}\right)$, $f(z)$ is the neuron activation function, z_i^{l+1} is the weighted sum of the input values of the *i-th* neuron in the $(l + 1)$-*th* layer, and its expression is:

$$z_i^{l+1} = \sum_{j=1}^{S_l} w_{ij}^{(l)} x_j + b_i^l, \tag{1}$$

where w_{ij}^l represents the weight coefficient connecting the *j-th* neuron of the *l-th* layer and the *i-th* neuron of the $(l + 1)$-*th* layer, $b_i^{(l)}$ represents the bias term of the $(i + 1)$-*th* neuron of the $(l + 1)$-*th* layer, S_l is the number of neurons in the *l-th* layer. The neuron activation function often uses the sigmoid function $f(z) = \frac{1}{1+e^{-z}}$ whose value range is [0, 1], or use the *tanh* function $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ whose value is [− 1, 1].

Adding a sparsity limit to the self-encoder, that is, at the same time, only some hidden layer nodes are 'active', thus obtaining a sparse self-encoder. Based on this, KL

dispersion is introduced to measure the similarity between the average activation output of a hidden layer node and the set sparsity ρ:

$$KL\left(\rho\middle\|\hat{\rho}_j\right) = \rho\,\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}, \tag{2}$$

where $\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j x^{(i)}\right]$, m is the number of training samples, $a_j x^{(i)}$ is the corresponding output of the *j-th* node of the hidden layer for i samples. In general we set the sparse factor ρ = 0.05 or 0.1. The larger the KL dispersion, the larger the difference between ρ and $\hat{\rho}_j$, and the KL dispersion equal to 0 means that the two are completely equal, that is, $\rho = \hat{\rho}_j$. Therefore, we can add KL dispersion as a regular term to the loss function to constrain the sparse rows of the entire self-encoder network:

$$J_{sparse}(W,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\|x_i - \hat{x}_i\|^2 + \frac{\lambda}{2}\sum_{l=1}^{n_{l-1}}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}} W_l^2 + \beta\sum_{j=1}^{S_2} KL\left(\rho\middle\|\hat{\rho}_j\right), \tag{3}$$

where λ is the weight decay constant; n_l is the number of neural network layers; β is the coefficient that controls the unit of the sparse constraint, and S_2 is the number of hidden layer units.

3.2 The Implementation of Emerging New Classification

Sparse Auto-encoder automatically learns features from unlabeled data and gives better characterization than raw data. Based on this, we use sparse autoencoder to learn features from the original signal of the normal device and send the trained model to the local device. Then the device identifies the test sample according to the model. If the characteristics of the sample have a significant error compared to the characteristics of the normal signal, the sample is considered to be a fault signal and sent to the superior diagnostic layer.

Assume $X = \{(x_i, y_i)\}_{i=1}^{n}$, $x_i = \{t_{ij}\}_{j=1}^{c}$ is the training sample data set, where x_i represents the *i-th* training data, $y_i \in Y = \{1\}$ is the x_i corresponding label (note that only the normal device data is included in X), the x_i feature dimension is c, t_{ij} represents the *j-th* feature of the data x_i. $\hat{X} = \{(\hat{x}_i, y_i)\}_{i=1}^{n}$, $\hat{x}_i = \{h_{ij}\}_{j=1}^{v}$ is the SAE output data set of the training sample, where $\hat{x}_i$ represents the *i-th* output data, the $\hat{x}_i$ feature dimension is v, and h_{ij} represents the *j-th* feature of the data x_i; $X' = \left\{\left(x_i', y_i'\right)\right\}_{i=1}^{m}$, $x_i' = \{t_{ij}'\}_{j=1}^{c}$ is the test sample data set, where x_i' represents the *i-th* test data, $y_i' \in Y = \{1, 2, \ldots, K\}(K > 1)$ is the corresponding label. Calculate the output feature center M_j of the normal class:

$$M_j = \frac{1}{n}\sum_{i}^{n} h_{ij} \tag{4}$$

The wave matrix D of the *i-th* data is constructed by calculating the Euclidean distance between h_{ij} and M_j:

$$d_{ij} = \left\| h_{ij} - M_j \right\| \tag{5}$$

Calculate the feature threshold matrix θ_j:

$$\theta_j = \frac{1}{n} \sum_{i}^{n} d_{ij} \tag{6}$$

$$If\ h'_{ij} - M_j > k\theta_j,\ y'_i \neq 1,\ else\ y'_i = 1$$

where k is the control threshold.

4 Fault Classification and Update Implementation

This section describes the model fault classification and update function. The fault classification of the model is responsible from the fault classification layer. This layer receives the fault data transmitted by the local monitoring layer. The data includes known faults and unknown faults. The fault classification layer is responsible for classifying known faults and identifying unknown faults. For the detected unknown faults, upload them to the decision-making layer, and the decision-making layer specifically identifies the unknown fault and makes an update decision.

4.1 Implementation of the Fault Classification Layer

Since many previous work showed that completely random trees have been successfully applied to the classification problem [13–15]. We improved the random forest model to implement the function of the fault classification layer.

Assume $X = \{(x_i, y_i)\}_{i=1}^{n}$ is the training sample data set, Where x_i represents the *i-th* training data, $y_i \in Y = \{1, 2, 3, \ldots, K\}$ is the corresponding label; $X' = \left\{ \left(x'_i, y'_i\right) \right\}_{i=1}^{m}$ is the test sample data set, where x'_i represents the *i-th* test data, $y'_i \in Y = \{1, 2, \ldots, M\}(M > K)$ is the corresponding label; Randomly selecting a plurality of sample data from the training set X, and then randomly selecting one feature value of the data to divide the sample data into two subtrees, repeating the above two steps to continuously construct the child node until the number of data of the child node reaches the upper limit, then the construction of one tree is complete. Let $A = (x_1, x_2, x_3, \ldots, x_m)$ be the training instances that fall into the same leaf node, then build the ball O for set A, The center of A is defined as: $c = \frac{1}{m} \sum_{x \in A} x$, the radius of the ball O centered on *c:* $r = dist(c, e)$, where e is the farthest example of the distance c in A, and the $label(\mathrm{O})$ is the label that appears the most in set A. During testing, the test instances that fell into these balls were known fault, and the test instances that fell outside the ball were unknown fault.

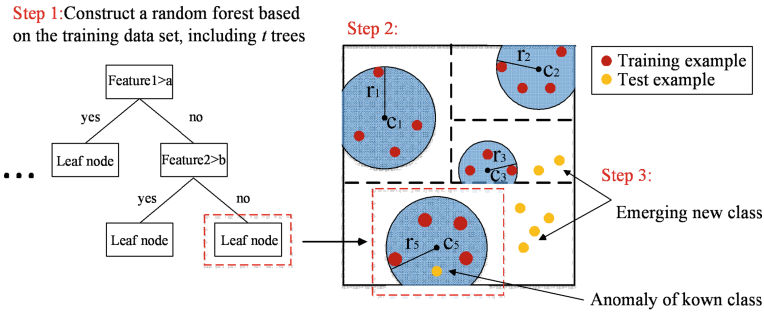

Fig. 3. The process of constructing and classifying a complete random tree.

Figure 3 shows the process of constructing and classifying a complete random tree:

Step1: Obtain a random forest model F according to the training data set X, which includes t random trees $\{f_1, f_2, f_3, \ldots, f_t\}$;

Step2: In each tree, build a ball O for an instance that falls into the same leaf node;

Step3: For test case x_i', x_i' is defined in the category tag of the *j-th* tree as:

$$label(x_i') = V_{\max}\left(\left\{f_j(x_i')\right\}_{j=1}^{t}\right),$$
$$f_j(x_i') = \begin{cases} label(\mathrm{O}) & \text{if } x_i' \text{ falls in a spherical area O} \\ new & else \end{cases}, \tag{7}$$

where $f_j(x_i')$ represents the label class calculated by the *j-th* tree for the input data x_i', which divides the x_i' into known or unknown classes. The function $V_{\max}\left(\left\{f_j(x_i')\right\}_{j=1}^{t}\right)$ indicates that the label category with the highest frequency of occurrence calculated by all trees is output. If $label(x_i') = new$, put the test instance into the buffer until the number of buffers reaches the upper limit and send it to the decision-making layer.

4.2 Decision-Making Layer Implementation

The unknown fault data uploaded by the fault classification layer may contain multiple unknown faults. The decision-making layer needs to detect it. If the data contains multiple unknown faults, correctly classify them, learn their characteristics and update the fault classification layer model.

For data set $S = \{x_i\}_{i=1}^{n}$ containing multiple unknown faults, we use the K-means clustering algorithm to cluster them. Since the K-means clustering algorithm requires the user to specify the number of clusters, the number of clusters that are too large or too small will make the data classification unreliable. Therefore, finding the optimal number of clusters is the key to achieving classification.

We use the Silhouette Coefficient to determine the number of clusters. The Silhouette Coefficient is a cluster validity index that combines the degree of agglomeration a_i and the degree of separation b_i, where b_i is the average distance of all other instances in the cluster to which the *i-th* instance belongs, and b_i is the average distance from the *i-th* instance to all instances in any cluster that does not contain the instance, then the Silhouette Coefficient of the *i-th* instance is $s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$, the value is [−1, 1], the larger the value, the better the clustering effect.

The specific classification processes are as follows:

(1) Determine the number of clusters as $N = 1 - \sigma\sqrt{n}$, n is the number of cluster data, σ is the control threshold.
(2) For the *N-th* cluster, calculate the contour coefficient of each data, and obtain $s(N) = \frac{1}{n}\sum_{i=1}^{n} s_i$, then the optimal number of clusters is $bestn = \arg\max(s(N))$.
(3) Output k-means clustering results according to *bestn*.

5 Experimental Results and Analysis

This paper uses the fan end bearing vibration data released by the Case Western Reserve University Bearing Experimental Center as the fault data for model verification. The data sampling frequency is 12 kHz. Six data types included in the data, which are the normal state, the inner ring fault F1, the rolling element fault F2, the central fault of the outer ring corresponding to the load zone F3, the central orthogonal fault of the outer ring corresponding load zone F4, and the center opposite fault of the outer ring corresponding load zone F5.

From the original signal, we collected 500 segments for each fault, each segment contains 1024 data points. Therefore each type of signal is represented by 500 data, and a total of 3000 data are considered for analysis.

The size of the original data in the high dimensional space is 3000 × 1024.

For the local detection layer, we used a subset of 70% normal data for training to obtain the FM-SAE diagnostic model. We will have 30% normal data set and F1 as test set 1; 30% normal data set, F1 and F2 as test set 2; 30% normal data set, F1, F2 and F3 as test set 3; 30% normal data set, F1, F2, F3 and F4 as test set 4; 30% normal data set, F1, F2, F3, F4 and F5 as test set 5. Figure 4 shows that after learning the characteristics of normal data, the FM-SAE separately characterizes the normal data and the fault data in the test set. Figure 4(a) (b) (c) show the fluctuations of three kinds of raw data, Fig. 4 (e) shows the normal data characteristic characterized by FM-SAE, and Fig. 4(f) (g) are the two fault data features characterized by FM-SAE. For a more intuitive display and normal data, the characteristics are shown in red. We can see that after learning the normal data characteristics, FM-SAE can easily distinguish it from other faults because their characteristics are very different.

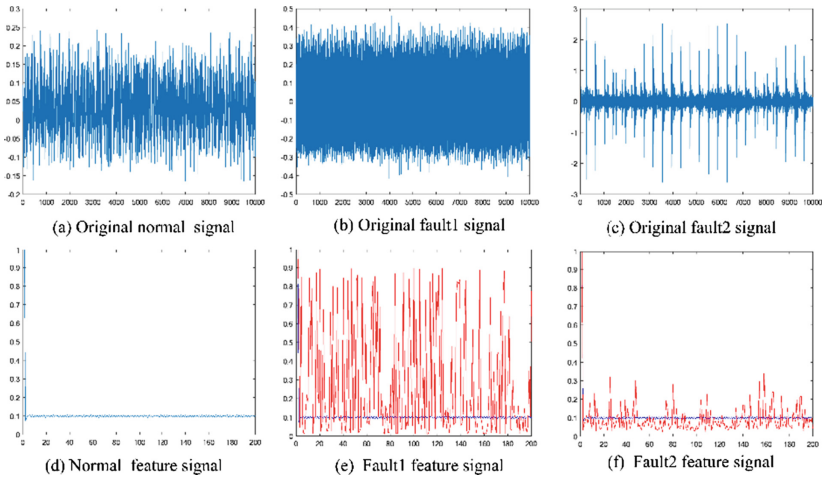

Fig. 4. Characterize normal data and fault data characteristics.

The network structure definition of FM-SAE consists of two hidden layers whose hidden layers contain 500 and 200 nodes, respectively. Figure 5 shows the identification fault accuracy curve with two methods (SAE and FM-SAE for monitoring anomalies). For SAE, it needs to learn the characteristics of all the faults to be classified, and classify it on this basis, which is not feasible in practical use. In the case of an increase in the number of faults, we can see that the classification accuracy is significantly reduced, because it is wrongly classified fault. FM-SAE only needs to learn normal data characteristics, and the classification accuracy remains basically the same when the type of failure increases.

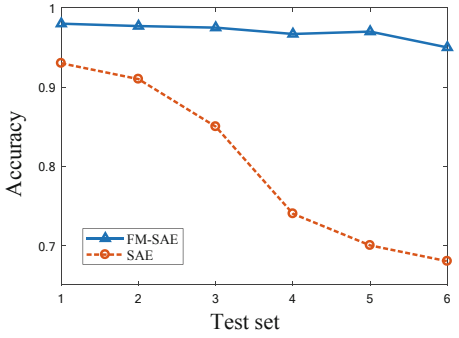

Fig. 5. Identification fault accuracy curve of two methods.

For the fault classification layer, we extract all the original fault data features, including the mean, standard deviation, peak, skewness, wave factor, crest factor and pulse factor, the feature data set size is 2500 × 6. Using 70% of F1, F2 fault characterization data for training to obtain a random forest model, leaving 30% of F1, F2 fault data and other fault data for testing. We will extract 70% F1, F2, F3 as the training set; extract the remaining 30% F1 as test set 1, remaining 30% F1 and all F4 as test set 2, remaining 30% F1 and all F4, F5 as test set 3, remaining 30% F1, F2 and all F4, F5 as test set 4, remaining 30% F1, F2, F3 and all F4, F5 as test set 4.

In order to verify the effectiveness of the improved random forest model classification in the fault classification layer, the effects of changing the number of trees and the number of fault types included in the unknown data set on the accuracy of fault classification detection are observed. During the experiment, other parameters are set unchanged. The experimental results are shown in Fig. 6. As shown in Fig. 6(a), the accuracy of the fault detection classification increases with the number of trees. When the number of trees exceeds 60, the accuracy of the fault detection classification increases slowly. From Fig. 6(b), in the case of five kinds of tests involving different types of faults, the fault classification accuracy rate is basically maintained above 92%.

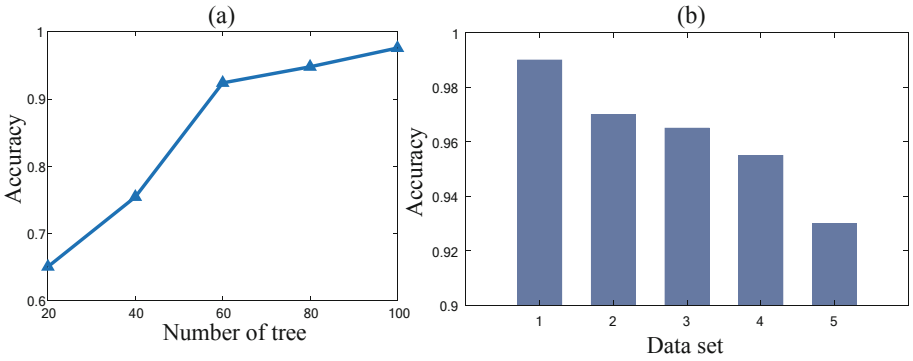

Fig. 6. Relationship between model parameter changes and fault classification accuracy.

For the decision-making layer, we use the three fault data sets F1, F2, and F3 to consider the case where N is equal to 1 to 20, cluster each N value and find the corresponding Silhouette Coefficient. Figure 7 shows the relationship between N and the Silhouette Coefficient. When the data is clustered into three categories, the Silhouette Coefficient is the highest.

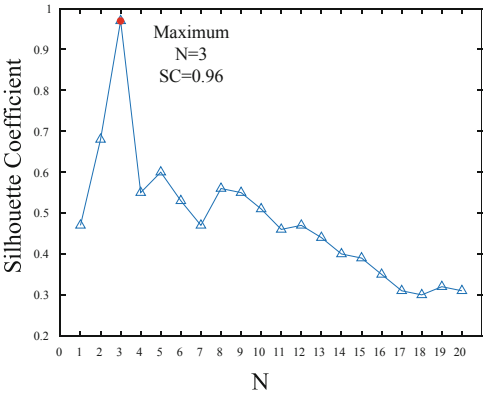

Fig. 7. The relationship between N and the Silhouette Coefficient.

6 Conclusion

In this paper, we propose a new distributed multi-level hierarchical fault diagnosis system based on SAE and random forest, called DFDS, for real-time monitoring of equipment. It can correctly classify faults and identify unknown faults while monitoring faults. Then, we verified the effectiveness of the system through experiments and evaluated the performance of each part to show the validity of the model. It provides new ideas for troubleshooting.

Acknowledgments. This work was supported by the State Grid Corporation Science and Technology Project (Contract No.: SG2NK00DWJS1800123).

References

1. Guoqian, J., Haibo, H., Jun, Y., et al.: Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox. IEEE Trans. Ind. Electr. **66**, 3196–3207 (2018)
2. Ince, T., et al.: Real-time motor fault detection by 1d convolutional neural networks. IEEE Trans. Ind. Electr. **63**, 7067–7075 (2016)
3. Tian, W.: Fault diagnosis of airborne equipment based on grey correlation fault tree identification method. In: World Congress on Intelligent Control & Automation IEEE (2008)
4. Yuliang, C., Tiejun, Z.: Research on the application of fuzzy fault tree analysis method in the machinery equipment fault diagnosis (2010)
5. Hong, G., Chen, X., Xue, X., et al.: Expert systems for fault diagnosis integrating neural network and fuzzy inference. In: International Conference on Information Technology (2011)
6. Lei, Z., et al.: Complex method of comprehensively evaluation and fault diagnosis in gun control system based on fuzzy reasoning. In: 11th International Conference on Electronic Measurement & Instruments. IEEE (2013)
7. Hu, Q., Zhang, R., Zhou, Y.: Transfer learning for short-term wind speed prediction with deep neural networks. Renewable Energy **85**, 83–95 (2016)

8. Chen, Z., Li, Z.: Research on fault diagnosis method of rotating machinery based on deep learning. In: 2017 Prognostics and System Health Management Conference (PHM-Harbin) (2017)
9. Xiuli, L., Xiaoli, X.: Fault diagnosis method of wind turbine gearbox based on deep belief network. Renewable Energy Res. (2017)
10. Qi, Y., Shen, C., Wang, D., et al.: Stacked sparse autoencoder-based deep network for fault diagnosis of rotating machinery. IEEE Access **5**, 15066–15079 (2017)
11. Chen, K., Hu, J., He, J.: Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder. IEEE Trans. Smart Grid **3**(9), 1748–1758 (2018)
12. Chen, Z., Li, W.: Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. IEEE Trans. Instrum. Measur. **66**, 1–10 (2017)
13. Yu, K., et al.: Classification with streaming features: an emerging-pattern mining ap-proach. ACM Trans. Knowl. Disc. Data **9**(4), 1–31 (2015)
14. Debarr, D., Ramanathan, V., Wechsler, H.: Phishing detection using traffic behavior, spectral clustering, and random forests. In: IEEE International Conference on Intelligence and Security Informatics IEEE (2013)
15. Zhang, M., Yian, L.: Signal sorting using teaching-learning-based optimization and random forest. In: 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), IEEE Computer Society (2018)
16. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)