Automatic Diagnosis of Microgrid Networks' Power Device Faults Based on Stacked Denoising Autoencoders and Adaptive Affinity Propagation Clustering

Xu, Fan; Shu, Xin; Zhang, Xiaodi; Fan, Bo

Published in:
Complexity

Published: 01/04/2021

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1155/2020/8509142

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 26/04/2021
Research Article

Automatic Diagnosis of Microgrid Networks’ Power Device Faults Based on Stacked Denoising Autoencoders and Adaptive Affinity Propagation Clustering

Fan Xu, Xin Shu, Xiaodi Zhang, and Bo Fan

1School of Data Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong 990777, China
2Wuhan Bridge Special Technology Company Limited of China Railway Major Bridge Engineering Group, Wuhan 730072, China
3State Grid Beijing Electric Power Company, Beijing 100080, China
4Office of Scientific Research and Development, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Bo Fan; fanbo@whu.edu.cn

Received 22 May 2020; Accepted 6 July 2020; Published 26 July 2020

Academic Editor: Ning Cai

Copyright © 2020 Fan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a model based on stacked denoising autoencoders (SDAEs) in deep learning and adaptive affinity propagation (adAP) for bearing fault diagnosis automatically. First, SDAEs are used to extract potential fault features and directly reduce their high dimension to 3. To prove that the feature extraction capability of SDAEs is better than stacked autoencoders (SAEs), principal component analysis (PCA) is employed to compare and reduce their dimension to 3, except for the final hidden layer. Hence, the extracted 3-dimensional features are chosen as the input for adAP cluster models. Compared with other traditional cluster methods, such as the Fuzzy C-mean (FCM), Gustafson–Kessel (GK), Gath–Geva (GG), and affinity propagation (AP), clustering algorithms can identify fault samples without cluster center number selection. However, AP needs to set two key parameters depending on manual experience—the damping factor and the bias parameter—before its calculation. To overcome this drawback, adAP is introduced in this paper. The adAP clustering algorithm can find the available parameters according to the fitness function automatic. Finally, the experimental results prove that SDAEs with adAP are better than other models, including SDAE-FCM/GK/GG according to the cluster assess index (Silhouette) and the classification error rate.

1. Introduction

As a key part of mechanical systems in the electric devices in microgrid networks, the operational health of bearings is related to the operation of the entire device [1–4]. Processing and analysis are an important basis for the evaluation of the health status of the electric devices in microgrid networks. Using vibration signals for fault diagnosis has become common in recent years.

For nonlinear and nonstationary signals, various feature extraction, and diagnosis methods are continuously developed; time and frequency indicators, wavelet transformation (WT), and empirical mode decomposition (EMD) are commonly used for fault feature extraction and have achieved significant results. However, various time-frequency domain indicators and wavelet transformation (WT) cannot adaptively decompose vibration signals because different vibration signals have different working frequency bands. Thus, EMD is proposed as a way to adaptively decompose the signal into intrinsic mode functions (IMFs) based on the current envelope mean of the signal [5]. To overcome the drawbacks, such as mode-mixing problems cause by noise in EMD, a model named “ensemble empirical mode decomposition” (EEMD) was first presented in [6]. Many scholars have already applied EEMD in fault diagnosis [7, 8]. However, these traditional methods depend highly on manual experience and prior knowledge, such as choosing the available time frequency indicators and wavelet basis function, and they also need to integrate several available models for fault feature extraction.
An increasing number of scholars have focused on deep learning in fault diagnosis due to its powerful automatic extracting features. For example, many studies have employed stacked autoencoders (SAEs) to extract features and fault diagnosis automatically and successfully [9–13]. However, most of these papers consider SAEs with a classifier and data labels to complete the fault diagnosis. However, the data obtained from different actual engineering platforms contain noise, such as voice data and vibration signals in practice engineering. To enhance the robustness of SAEs, stacked denoising autoencoders (SDAEs) were created [14, 15]. Compared with SAEs, SDAEs introduce artificial noise, and some of the input data are randomly zeroed to reconstruct the original input data. In addition, SDAEs have been widely applied in many domains [16–18]. Therefore, SDAEs are utilized to extract bearing fault characteristics directly from the frequency domain signal to reduce the manual experience dependence in this paper.

In addition, marking data labels requires a great deal of labor and richly experimental engineering when working with large amounts of data. Therefore, no manual experience or prior knowledge is required to mark the fault type and fault label by using SDAEs without an output layer.

To identify the different fault types automatically, cluster model is used to complete the fault diagnosis without data labels in this paper. Fuzzy C-mean (FCM) is a commonly used model in fault diagnosis [19]. In FCM, to compute the distance between samples, Euclidean distance is often employed; hence, it is only available for data with a spherical distribution. Actually, many datasets do not have such characteristics. To solve this problem, the Gustafson–Kessel (GK) clustering algorithm [20] features an objective function based on the covariance matrix and is suitable for the cluster analysis of datasets with the correlation between variables [21]. However, the FCM and GK methods are still aimed at datasets with spherical shapes, while the Gath–Geva (GG) clustering method computes the distance between any two adjacent data points by using the maximum likelihood distance and has been successfully applied to the diagnosis of rolling bearing faults [22, 23]. In [24, 25], the authors used EEMD and GG to complete the bearing fault diagnosis.

However, all of the clustering models mentioned above need to preset the cluster numbers through manual experience before calculation. The affinity propagation (AP) clustering algorithm can automatically find the appropriate number of clusters. AP continuously performs message passing and iterative looping to generate K high-quality clusters, and it uses energy functions to minimize clusters and assign each data point to the nearest cluster [26]. Actually, there are two key parameters—the bias parameter \( p \) and damping factor \( \lambda_m \)—in the AP cluster model. One researcher [26] recommended setting \( p \) to the median value of all \( p_m \) samples without prior knowledge at the beginning stage. However, sometimes \( p_m \) cannot induce the AP algorithm to generate available cluster numbers because \( p_m \) is not selected on the basis of the clustering structure of the dataset itself. When the AP algorithm oscillates (i.e., the number of clustering classes generated during the iteration is constantly oscillating) and cannot converge, increasing \( \lambda_m \) can eliminate the oscillation, but \( \lambda_m \) must be manually increased when oscillating, and the algorithm is run again until the algorithm converges. The approach is to set \( \lambda_m \) directly to close to 1 to avoid oscillation, but the reliability and availability updates are slow, and the algorithm runs slowly. Therefore, Wang et al. developed a model named adaptive affinity propagation (adAP) to find the best cluster according to the cluster assessment index (Silhouette) [27]. The adAP scan biases parameter space to look for the cluster number space to find the suitable clusters, and it adjusts the damping factor to weaken the oscillation and adaptive escape oscillating technique when the damping factor method fails.

Therefore, a method based on SDAEs and adAP for bearing fault diagnosis is presented in this study. The main attributes are presented in the following section:

1. Different from traditional multistep fusion fault diagnosis methods and the basic SDAE model, which require data labels for fault classification, SDAEs without an output layer are utilized to extract fault features directly from the frequency domain and weaken the dependence on manual experience to mark the data labels in this paper.

2. There are few reports in the literature in which the adAP model is applied to bearing fault diagnosis.

3. To prove the extracting feature performance of the proposed model (SDAE-adAP), classification accuracy and Silhouette are used to demonstrate the performance of adAP in suppressing some other models, such as FCM/GK/GG.

The rest of this paper is organized as follows. Section 2 contains a review of SAE. Experiment data and detailed procedures are presented in Section 3. A comparison analysis of the experiments is described in Section 4, and Section 5 concludes the paper.

2. Review of the SDAEs and adAP

2.1. Stacked Denoising Autoencoders

2.1.1. Autoencoders. Autoencoders (AEs) include encoders and decoders [28]. The basic structure is shown in Figure 1.

Encoders are used to map the input to the following hidden layer and obtain a new nonlinear extracted hidden feature \( z \) by using the following equation:

\[
  z = f(x) = s(w^{(1)}x + b),
\]

where \( x' = [x_1, x_2, x_3, \ldots, x_j, \ldots, x_n], i = 1, 2, 3, \ldots, N, j = 1, 2, 3, \ldots, n, N \) is the sample number, \( n \) denotes the length of each sample, \( w^{(1)} \) represents the connection matrix used to connect the original input data and the \( L \)th hidden layer, \( s \) signifies the sigmoid active function \( s(x) = 1/1 + e^{-x} \), and \( b \) is the bias item.

The decoder is utilized to map and reconstruct the extracted hidden feature \( z \) close to the original input \( x \). The procedure above is calculated by
where $g$ is the sigmoid function. The construction error is calculated by

$$J = \|x - g(f(x))\|^2,$$

and $J$ can replace the following cost function:

$$J = \|x - g(f(x))\|^2 = J(w, b)$$

$$= \left[ \frac{1}{N} \sum_{i=1}^{N} \|x^{(i)} - g(f(x^{(i)}))\|^2 \right] + \lambda \sum_{L=1}^{L-1} \sum_{j=1}^{r_L} (w_{ji}^{(L)})^2,$$

where $r_L$ is the neural node number at the $L^{th}$ hidden layer; $\lambda$ is a regularization coefficient.

2.1.2. Denoising Autoencoders. Denoising autoencoders (DAEs) mix the training data into the noise (the data are randomly set to zero) and remove the noise to obtain the reconstructed output data. In the case of destroyed data, DAEs achieve a better description of the input data and enhance the robustness of the entire model. The structure of a DAE is shown in Figure 2.

In Figure 2, $x$ indicates the raw input data and $x_1$ represents the destroyed input data according to the denoising rate $P$, $y$ is the extracted feature obtained from $x_1$ by using the sigmoid function, and $z$ denotes the output. The difference between DAEs and AEs is that DAEs destroy input data by denoising the rate $P$ to $x \rightarrow P(x_1/x)$. The reconstructed error between the output $z$ and the original input $x_1$ is

$$J_D = \|x - g(f(x_1))\|^2 = J_D(w, b)$$

Hence, the cost function in equations (4) and (5) can be rewritten as

$$J_D = \|x - g(f(x_1))\|^2 = J_D(w, b)$$

$$= \left[ \frac{1}{N} \sum_{i=1}^{N} \|x^{(i)} - g(f(x^{(i)}))\|^2 \right] + \lambda \sum_{L=1}^{L-1} \sum_{j=1}^{r_L} (w_{ji}^{(L)})^2.$$

2.1.3. Stacked Denoising Autoencoders. The SDAE concept was presented by Vincent [14, 15]. The core idea of SDAE is adding noise to the input data of each encoder. Therefore, a more robust feature expression can be learned. Figure 3 shows the structure of an SDAE. The learning process of SDAEs can be divided into two steps.

The first is the greedy layer-by-layer learning of SDAEs using unmarked samples. The specific process is as follows: assuming that the total number of hidden layers is $L$, input the original data into the first layer of the DAE, and perform unsupervised training to obtain the parameter $W(1)$ of the first hidden layer. After each step, the trained $(L - 1)^{th}$ layer is selected as input to train the $L^{th}$ hidden layer to obtain $W(L)$, and the weights of each layer are trained. Second, the reconstructed error is reduced by the backpropagation method, which is also utilized to update the parameters and to make the network converge.

In the backpropagation error calculation process, it is necessary to calculate the residual $\delta$ of each hidden layer. For each output node $i$, $\delta$ is calculated as follows:

$$\delta_i^{L-1} = a_i^{L-1}(1 - a_i^{L-1}) \sum_{j=1}^{r_L} w_{ji}^{L-1} \delta_j^L,$$

where $a_i^L$ denotes the output at the $L^{th}$ hidden layer.

Use equations (8) and (9) for the SDAE network.

To adjust the parameters of each hidden layer, use the following equation:

$$w_{ji}^L = w_{ji}^{L-1} - \beta a_i^{L-1} \delta_j^L,$$

$$b_i^L = b_i^{L-1} - \beta \delta_i^L,$$

where $\beta$ is the learning rate.

It should be mentioned that the input $x$ is normalized before SAE and SDAE training; hence, the output range at each hidden layer should be [0, 1]. Moreover, the output range of the sigmoid function is [0, 1]. The sigmoid function’s curve changes continuously between [0, 1]. Therefore, we chose the sigmoid function as the active function in this paper. In addition, the reconstruction error is not calculated for all training data; rather, in each iteration, the reconstruction error of certain training data is randomly optimized by the stochastic gradient descent model. Hence, the update speed for each round of parameters is greatly accelerated. Therefore, the gradient descent optimization model is used to update the weight parameter $w$ and bias item $b$ in this paper.
2.2. adAP Clustering Model. The AP algorithm works on the \( N \times N \) similarity matrix \( S \) composed of \( N \) data points and regards all samples as cluster center point candidates at the beginning stage [26]. There are some tight clusters in the feature space, and the function \( E(C) \) of one cluster represents the similarity sum between any one data point and the cluster centers. The calculation of \( E(C) \) is as follows:

\[
E(C) = -\sum_{j=1}^{K} S_j(i, C_i),
\]

(10)

where \( K \) denotes the cluster number, \( C_i \) denotes the \( i \)th cluster center, and \( S(i, C_j) \) represents the distance between each point and the corresponding cluster center point. The negative value of the distance between the two any adjacent points is assumed to be the degree of attraction or attribution, the \( k \)th point is more attractive to the closer data \( i \)th point, and data point \( i \) agrees that the \( k \)th point has a greater sense of belonging to its cluster center. Therefore, the \( k \)th point of the cluster center is more attractive to other data points, and the possibility of becoming a cluster center will become greater.

The AP algorithm continuously collects relevant evidence from the data to select the available class representation: AP uses \( R(i, k) \) to describe the degree to which the data point \( k \) is suitable as the cluster center point of data point \( i \). \( A(i, k) \) is called the degree of attribution and is used to describe the extent to which data point \( i \) selects data point \( k \) as its cluster center point. The literature [27] shows that the larger the \( R \) and \( A \) values of the data point \( k \), the greater the probability will be that the data point \( k \) becomes the cluster center.

The AP algorithm generates \( k \) high-quality cluster classes through an iterative loop and minimizes the energy function of the cluster class. Finally, it assigns each data point to the nearest cluster class.

There are two key parameters (i.e., the bias parameter \( p \) and the damping factor \( \lambda \)) in AP. The deviation parameter \( p(i) \) (usually a negative number) represents the degree to which the data point \( i \) becomes the cluster center.

As mentioned above, \( R(i, k) \) and \( A(i, k) \) can be calculated by

\[
R(i, k) = S(i, k) - \max[A(i, j) + S(i, j)],
\]

\[
( j = 1, 2, 3, \ldots, N, j \neq k, j \neq i),
\]

(11)

\[
A(i, k) = \min[0, R(k, k) + \sum \{\max(0, R(j, k))\}],
\]

\[
( j = 1, 2, 3, \ldots, N, j \neq k, j \neq i).
\]

(12)

From equations (11) and (12), when \( p(k) \) is large, \( R(k, k) \) and \( A(i, k) \) also become larger; hence, the class represents \( k \) as the final cluster center. When \( p(i) \) is larger, more cluster classes represent the final cluster center. Therefore, increasing or decreasing \( p \) affects the number of clustered classes. The authors recommend setting all \( p \) to \( p_m \) (which represents the median value of all elements in \( S \) without prior knowledge at the beginning stage [26]. However, in many cases, \( p_m \) cannot make the AP algorithm produce optimal clustering results because the setting of \( p_m \) is not based on the clustering structure of the dataset itself. When the AP algorithm oscillates (i.e., the number of clustering classes generated during the iteration is constantly oscillating) and cannot converge, increasing \( \lambda \) can eliminate the oscillation. When oscillating, one must manually increase \( \lambda \) and rerun the algorithm until the algorithm converges. The approach is to set \( \lambda \) directly close to 1 to avoid oscillation, but the \( R(i, k) \) and \( A(i, k) \) updates are slow, and the algorithm runs slowly.

To overcome the drawbacks mentioned above, adAP searches the cluster number space by scanning the bias parameter space to find the optimal clustering result (called adaptive scanning) and adjusts the damping factor \( \lambda \) to eliminate the oscillation (called adaptive damping), thus lowering the \( p \) value to escape concussion (called adaptive escape).

The goal of adAP is to eliminate both oscillating and fast algorithms when oscillation occurs. Although it is more likely to increase \( \lambda \) to near 1 to eliminate oscillation, the larger \( \lambda \) is, the slower the \( R \) and \( A \) in equations (11) and (12) and the update become, and the more iterations the algorithm needs to achieve the update effect when \( \lambda \) starts from 0.6.

The adaptive adjustment damping factor technique is designed as follows:

1. The AP algorithm performs a loop to detect whether oscillation is occurring.
2. If there is oscillation, increase \( \lambda \) by one step (for example, 0.05); otherwise, proceed to step 1.
3. Continue \( w \) cycles (to see the effect after \( w \) cycles).
4. Repeat the steps above until the algorithm reaches the stop condition.

Complexity
If increasing \( \text{lam} \) (e.g., \( \text{lam} \) is increased to 0.85 or higher) fails to depress oscillations, an adaptive escape technique should be designed to avoid oscillations. The fact that large \( \text{lam} \) has little effect suggests that oscillations are persistent under the given \( p \), so the alternative is to decrease \( p \) away from the given value to escape from oscillations. This escape method is reasonable since it works together with the adaptive scanning of \( p \) discussed below, different from AP, which works under a fixed \( p \). The adaptive escape technique is designed as follows: when oscillations occur and \( \text{lam} \geq 0.85 \) in the iterative process, \( p \) is decreased gradually until oscillations disappear. This technique is added in step 2 of the adaptive damping method: if oscillations occur, increase \( \text{lam} \) by a step (e.g., 0.05); if \( \text{lam} \geq 0.85 \), decrease \( p \) by step \( p_{\text{step}} \). Otherwise, go to step 1 of the adaptive damping method. Both adaptive damping and adaptive escape techniques are used to eliminate oscillations at the same time. The monitoring window size \( w \) = 40 is appropriate as per our experiences (but occasionally, random vibrations and tolerant vibrations in initial iterations will be caught under too small \( w \), and AP runs slowly under a \( w \) value that is too large). The pseudocodes of adaptive damping and adaptive escape are shown in the work of Kan et al. [28] (and \( \text{maxits} \) and \( p_{\text{s}} \) will be set in the following step).

To keep the algorithm fast, the design of the bias parameter \( p \) is as follows:

The algorithm starts from the initial given \( p \), and each iteration of the cyclic process updates the \( R \) and \( A \) (but the similarity matrix \( S \) is fixed); if the cyclic process converges to a certain cluster number \( K \), in the stride \( p_{\text{step}} \) manner, gradually reduce \( p \)—that is, change \( p(i) \) on the diagonal of \( S \)—and repeat the same cyclic process to obtain a different \( K \). To avoid double counting, use the current \( R \) and \( A \) values after each reduction of \( p \) as a new starting point and continue to calculate \( R \) and \( A \). The adaptive scanning technology for \( p \) is designed as explained in this section.

The acceleration technology of the \( p \) drop is designed as follows:

1. The AP algorithm performs an iteration to check whether the number of cluster classes converges to \( K \). If yes, go to step 2. Otherwise, \( b = 0 \); repeat step 1.

2. Check if the number of clustering classes converges to \( K \) and \( b < \text{iter}_{\text{max}} \); if yes, count \( b = b + 1 \).

Otherwise, go to step 1.

If (3) \( p = p + b \times p_{\text{step}} \), go step 2.

The pseudocodes of adaptive \( p \)-scanning technology are shown in reference [28].

3. Experiment Setup

3.1. Experiment Data. The four basic faults—that is, normal (NR), ball fault (BF), inner race fault (IRF), and outer race fault (ORF)—were collected from a motor driving a power device [29]. The sampling frequency is 12 KHz. The fault diameters are 0.18 mm, 0.36 mm, and 0.54 mm. Detailed information about the data is displayed in Table 1. Each sample contains 2,048 points, and 50 samples are included under different conditions; thus, 500 samples were used for training and testing. Note that two datasets (A and B) were used for this study. Each dataset contains nine fault types under different working conditions. For each dataset, 300 samples under various conditions were chosen as the training dataset and the remaining 200 samples constitute the testing sample.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Fault diameters (mm)</th>
<th>Fault type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B</td>
<td>0.18/0.36/0.54</td>
<td>NR, BF1, IRF1, ORF1, BF2, IRF2, ORF2, BF3, IRF3, ORF3</td>
</tr>
</tbody>
</table>

3.2. Evaluation Index. By searching the class number space, adAP can output some clustering results with various cluster numbers. Therefore, the clustering validity method can be used to assess the performance of the clustering results. Among the many effectiveness indicators, the Silhouette index is widely used because of its evaluation ability for obvious cluster structures. The Silhouette index shows the interclass tightness of the cluster structure and the class separability [30]. Therefore, the Silhouette index is taken as an example to solve the optimal clustering result.

A dataset is divided into \( K \) clusters \( C_i \) (\( i = 1, 2, \ldots, K \)), \( a(t) \) is the mean distance between the sample \( t \) and other samples in the cluster \( C_i \), and \( d(t, C_i) \) represents the average distance between sample \( t \) and all of the samples in the cluster \( C_i \). Then, \( b(t) = \min \{ d(t, C_i) \} i = 1, 2, 3, \ldots, K \). Therefore, the Silhouette index of a sample is calculated by

\[
S_{\text{t}}(t) = \frac{b(t) - a(t)}{\max[a(t), b(t)]}
\]  

It is easy to calculate the average \( S_{\text{av}}(C_i) \) of all samples of a cluster \( C_i \). It reflects the tightness of the cluster \( C_i \) (such as the average distance within the cluster) and the separability (such as the minimum interclass distance). The average value \( S_{\text{av}}(C) \) for overall samples by using \( S_{\text{t}} \) can reflect the quality of the clustering results.

For a series of Silhouette index values of clustering results, the larger the value is, the better the clustering quality becomes. The cluster number corresponding to the largest value is the optimal cluster number, and the corresponding clustering result is also optimal [31]. The Silhouette value of the clustering result exceeding 0.5 denotes that each cluster can be separated well [31].

3.3. Procedures for the Proposed Model. The detailed procedures of the proposed method contain three sections: (1) data preprocessing, (2) feature extraction, and (3) fault diagnosis.
(1) Data preprocessing: the fast Fourier transformation (FFT) is utilized to transform the raw signal from the time domain to the frequency domain because the coefficient matrix is symmetrical after FFT operation. Hence, half of the coefficient matrix was used for SAE and SDAE training and testing. All input data are normalized into [0, 1].

(2) Feature extraction: since the dimension of the original data is high and cannot be visualized, PCA is used to reduce the feature dimension at each hidden layer to compare the feature extraction performance of the SAE and SDAE. Note that the extracted feature vector dimension is 3 at the final hidden layer without PCA operation.

(3) Fault diagnosis: after training the SAE and SDAE, 3-dimensional feature vectors are considered as the inputs of FCM, GK, GG, and adAP for fault identification. To verify that the clustering performance of the proposed SDAE-adAP is better than other models, such as SAE/SDAE-FCM/GK/GG, SAE-adAP, a clustering evaluation index (i.e., Silhouette) is used to assess the clustering performance results. In addition, the accuracy is also utilized to compare the identification performance of the different models. The detailed procedures are shown in Figure 4.

4. Fault Diagnosis and Comparison Analysis

4.1. Feature Extraction for Different Vibration Signals

First, the origin vibration signal is shown in Figure 5. As seen in Figure 5, distinguishing all signals is not easy. BF, IRF, and ORF signals have regularity, while all NR signals have no obvious periodic regularity because they are random vibrations, and their self-similarity is poor. Different from NR signals, BF, IRF, and ORF vibration signals contain a fixed vibration period in some unique frequency bands, and the self-similarity is higher than in NR signals. Particularly, when the inner ring is fixed, the outer ring rotates with the bearings, and the vibration regularity in BF signals becomes clearer. Therefore, the BF, IRF, and ORF vibration signals have strong periodic regularity, but it is still not easy to identify these fault vibration signals. To extract the useful fault feature effectively and identify these different signals easily, the FFT is utilized to transform the vibration signal because the frequency domain signal contains useful fault information [9]; here, take a BF2 signal as an example. The FFT result of the BF2 is shown in Figure 6. As illustrated in the two subfigures on the right in Figure 6, the working frequencies for BF primary highlight form 0 Hz to 150 Hz, as the BF signal working frequency is 58 Hz; thus, the fault frequency is a primary highlight on 58 Hz and the double frequency (117.2 Hz). Therefore, these results prove that the frequency domain signal subsequently contains useful fault information.

The coefficient matrixes are used for feature extraction through eight hidden layers. Some parameters in the SAE and SDAE should be set before training, such as the input size, the learning rate, the denoising rate, and the total number of the neural nodes at each hidden layer.

The length of each original sample is 2,048 points. The frequency domain coefficients of each sample after FFT transformation are symmetrical; hence, the length of each input sample in the SDAE is transformed to 1,024. In addition, the hidden layer adopts a triangular structure—that is, the number of nodes in the following adjacent hidden layer is half that of the previous hidden layer. Therefore, the number of nodes in the first hidden layer is 512. The neural node numbers at the first eight hidden layers are selected as 512, 256, 128, 64, 32, 16, 8, and 3. Then, the first three principal components (PCs) in PCA are chosen as the fault feature for data visualization and compared to the feature extraction ability of the SAE and SDAE.

Since much information is missing when the denoising probability \( p \) becomes too large, the SDAE will generate a high error rate. The authors suggest that parameter \( p \) is often set lower than 0.5 [32, 33]. \( p \) is set as 0.15 in this paper.

If the learning rate is too high, the convergence speed of reconstruction error will be fast, but it is easy to trap into the local optimal point. However, if the learning rate is too small, the SAE and SDAE models will exhibit slow convergence [34–41]. The learning rate \( \beta \) in equation (8) is 0.1, and the largest iteration number is 3,000 in this study.

The 3-dimensional results of different datasets for the training dataset through eight hidden layers by using SDAE/SAE with PCA dimension reduction under different conditions are shown in Figures 7 and 8. In Figure 7, “SAE-A-512-training data” means that 512 neural nodes exist at the first hidden layer through the SAE and dataset A. As shown in Figure 8, the various fault samples are separated well when the number of hidden layers increases.

In Figure 8, to the naked eye, the last two subfigures show only a shape, such as “BF3.” However, in the former first seven hidden layers, compared with the SAE, when the number of hidden layers increases, the SDAE feature extraction ability becomes stronger. As with the SDAE-A-2 training data, all the BF2 samples look like only one sample at the naked-eye layer.

4.2. Fault Diagnosis by Using the adAP-Training Dataset

The results of adAP clustering are shown in Figure 9. The choice of descending stride \( p_{\text{step}} \) is the key to adAP. The smaller the \( p_{\text{step}} \) is, the more slowly the algorithm runs. Conversely, the larger the \( p_{\text{step}} \) is, the more likely it becomes that the number of clusters reflecting the intrinsic cluster structure of the dataset will be missed. The fixed stride is difficult to adapt to different cases of large and small clusters.

Therefore, the adaptive adjustment technique of the descending stride \( p_{\text{step}} \) is as follows:

\[
p_{\text{step}} = 0.01 p_m \frac{q}{K + 50},
\]

where \( q = 0.1 \sqrt{K + 50} \). Hence, the algorithm can dynamically adjust \( q \) when generating \( K \) clusters to achieve a smaller step size when \( K \) is larger or a smaller step when \( K \) is smaller.

When clustering \( N \) data points, it is generally considered...
Before training SAE and SDAE, initializing the parameters in SAE and SDAE models

Before fault diagnosis, initializing the parameters in FCM, GK, GG, and adaptive AP models

Figure 4: The procedure of the proposed model.

Figure 5: Continued.
reasonable that the upper limit of the optimal number of clusters is the square root of $N$ [35]. When the initial $p = \frac{p_m}{2}$, the number of cluster classes $K$ that the algorithm converges at first can basically reach or exceed $\sqrt{N}$. However, the cluster number searched by the AP algorithm is more than $\sqrt{N}$ (because each data point is viewed at the beginning of the algorithm). For the cluster class, the starting value can be set to $p = \frac{p_m}{2}$. The minimum number of cluster 2 determines the lower bound of $p$, reducing $p$ until the cluster number $K = 2$. To prevent the maximum number of
Figure 7: Continued.
iterations from affecting whether the algorithm reaches $K = 2$, the largest iteration number $\text{iter}_{\text{max}}$ is fixed as 50,000 in this study.

After the parameters mentioned above are preconfigured, 3-dimensional features are chosen as the input of adAP for fault diagnosis. The 3-dimensional clustering results for training datasets A and B by using an SAE/SDAE with adAP are shown in Figure 9.

1. In Figure 9, symbol "CC" denotes the cluster center points. There are two cluster points for BF2 samples, and the diamond symbol indicates the BF2 sample using dataset A. In the third subfigure, all BF2 samples have only one cluster center point.

2. From Figure 9, all samples in the third and fourth subfigures are separated well, and they are very close to its cluster center point. The division between each cluster class is very clear. For example, BF3 and IRF3 samples are separated well by using an SDAE for dataset B in the fourth subfigure in Figure 9. Therefore, these scattered points easily lead to the generation of multiple or extra cluster center points.

These results demonstrate that the robustness and the feature extraction ability of SDAEs are better than those of SAEs. Moreover, adAP can find the cluster center point automatically.

The result of the energy function $E(C)$ by using an SDAE for training dataset A is shown in Figure 10. As seen in Figure 10, there are obvious oscillations in the curve during the first 130 iterations. Increasing the value of $\text{lam}$ in the following steps gradually keeps the curve stable, as evidenced by the fact that the largest value of $E(C)$ occurs when the cluster number is 9. Actually, the parameter $\text{lam}$ increases up to 0.7 when the number of iterations is 101, but the curve also has a random oscillation. Then, $\text{lam}$ increases to 0.75
Figure 8: Continued.
when the curve becomes stable starting from 131. With the increment of the iteration number, the value of \( \text{lam} \) becomes smaller than the former iteration. Hence, the best cluster number is 9. The clustering index (Silhouette) in equation (13) is also used to assess the clustering result. The results of the Silhouette index with different cluster numbers by using an SAE/SDAE with adAP (the training dataset) are displayed in Table 2.

(1) From Table 2, the largest value is 0.954 by using an SDAE with dataset A when the cluster number is 9, but for the SAE used in dataset A, the largest value is 0.694, which is smaller than 0.954. It should be noted that the cluster number is 10, not 9, because the SAE generated some scatter points, such as BF3 in Figure 9. This result leads to the generation of extra cluster center points for the same fault samples.

(2) Although the best cluster number is 9 in the SAE, it is the same in the SDAE for dataset B, while the largest value of the Silhouette index is 0.889. Hence, the feature extraction ability of the SDAE exceeds that of the SAE, and adAP can find the available parameters automatically. The classification accuracy using the best cluster number is shown in Table 3. The lowest classification error rate is 0% for dataset A by using the SDAE, and the classification error rate of the SDAE is lower than that of the SAE on the whole.

4.3. Compared with FCM, GK, and GG. To further demonstrate that the proposed model (SDAE-adAP) is better than SAE/SDAE-FCM/GK/GG, the 3-dimensional clustering results for training dataset A and B by using the SAE/SDAE with FCM/GK/GG are shown in Figures 11 and 12. Compared with the SAE, most of the samples are separated well and close to the center point in the SDAE. While some samples exhibit an overlap phenomenon—especially

![Figure 8: The 3-dimensional results of different datasets for the training dataset through eight hidden layers by using an SDAE with PCA dimension reduction; 1–8 denote the hidden layer number. (a) SDAE-A-512-training data. (b) SDAE-B-512-training data. (c) SDAE-A-256-training data. (d) SDAE-B-256-training data. (e) SDAE-A-128-training data. (f) SDAE-B-128-training data. (g) SDAE-A-64-training data. (h) SDAE-B-64-training data. (i) SDAE-A-32-training data. (j) SDAE-B-32-training data. (k) SDAE-A-16-training data. (l) SDAE-B-16-training data. (m) SDAE-A-8-training data. (n) SDAE-B-8-training data. (o) SDAE-A-3-training data. (p) SDAE-B-3-training data.](image-url)
IRF1 and BF1 in Figure 11(a)—these samples are separated well by using an SDAE. The classification accuracy achieved by using different combination models for training datasets A and B are displayed in Table 4. The lowest error rate is 0% in dataset A, and the lowest error rate of the proposed model SDAE-adAP is lower than other combination models, including SAE/SDAE-FCM/GK/GG and SAE-adAP.

4.4. Fault Diagnosis through the adAP-Testing Dataset.

The testing datasets is used to test the performance model. As with the training dataset, the feature extraction procedure through several hidden layers in the SAE and SDAE is shown in Figures 13 and 14. As seen in Figures 13 and 14, all of the testing samples are separated well at the final hidden layer, such as the ORF2 samples. All of the ORF2 samples look like
Figure 10: The results of the energy function (similarity) for all samples.

Table 2: The results of the Silhouette index with different cluster numbers by using an SAE/SDAE with adAP (the training dataset). Here, “—” means there is no cluster number equal to 10, 11, or 12.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Silhouette index</th>
<th>Cluster no.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 3 4 5 6 7 8 9 10 11 12</td>
</tr>
<tr>
<td>SAE-adAP</td>
<td>A</td>
<td>0.330 0.391 0.484 0.538 0.592 0.635 0.664 0.683 0.694 0.672 0.662</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.483 0.510 0.636 0.608 0.630 0.574 0.641 0.678 — — —</td>
<td></td>
</tr>
<tr>
<td>SDAE-adAP</td>
<td>A</td>
<td>0.413 0.549 0.695 0.738 0.808 0.839 0.935 0.954 — — —</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.431 0.512 0.602 0.713 0.812 0.875 0.864 0.889 — — —</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The results of the best cluster number and classification error accuracy at the maximum Silhouette index value by using different models (the training dataset).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Best cluster no.</th>
<th>Silhouette index</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>SAE-adAP</td>
<td>10</td>
<td>0.6945</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>SDAE-adAP</td>
<td>9</td>
<td>0.9545</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SAE-adAP</td>
<td>9</td>
<td>0.6785</td>
<td>4.07</td>
</tr>
<tr>
<td>B</td>
<td>SDAE-adAP</td>
<td>9</td>
<td>0.8898</td>
<td>2.22</td>
</tr>
</tbody>
</table>

Figure 11: Continued.

FIGURE 12: Continued.
Figure 12: The 3-dimensional clustering results for training dataset B by using an SAE/SDAE with FCM/GK/GG. (a) SAE-FCM-B. (b) SAE-GK-B. (c) SAE-GG-B. (d) SDAE-FCM-B. (e) SDAE-GK-B. (f) SDAE-GG-B.

Table 4: The error classification accuracy obtained through various models (the training dataset).

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FCM</td>
</tr>
<tr>
<td>SAE</td>
<td>A</td>
<td>8.15</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>2.59</td>
</tr>
<tr>
<td>SDAE</td>
<td>A</td>
<td>11.1</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>12.9</td>
</tr>
</tbody>
</table>

Figure 13: Continued.
Figure 13: Continued.
Figure 13: The 3-dimensional results of different datasets for the testing dataset through eight hidden layers by using an SAE with PCA dimension reduction; 1–8 denote the hidden layer number. (a) SAE-A-512-testing data. (b) SAE-B-512-testing data. (c) SAE-A-256-testing data. (d) SAE-B-256-testing data. (e) SAE-A-128-testing data. (f) SAE-B-128-testing data. (g) SAE-A-64-testing data. (h) SAE-B-64-testing data. (i) SAE-A-32-testing data. (j) SAE-B-32-testing data. (k) SAE-A-16-testing data. (l) SAE-B-16-testing data. (m) SAE-A-8-testing data. (n) SAE-B-8-testing data. (o) SAE-A-3-testing data. (p) SAE-B-3-testing data.

Figure 14: Continued.
a square at first glance by using the SDAE in Figure 14, while they are scattered in the SAE in Figure 13. Particularly in Figure 14, all samples, including the NR and other fault samples, are separated well when the number of hidden layers increases. The next step is to choose the extracted 3-dimensional features as inputs of adAP for fault diagnosis.

The 3-dimensional clustering results for the testing dataset by using the SAE/SDAE with adAP are shown in Figure 15. As seen in Figure 15, there are 10 cluster center points (red square points), but the actual number of clusters is 9, not 10. Moreover, all samples are separated well in the SDAE and scattered obviously around its cluster center point, as they were in Figures 13 and 14. The corresponding results of the Silhouette index with different cluster numbers by using the SAE/SDAE with adAP (the testing dataset) are listed in Table 5. In Table 5, the largest value of the Silhouette index values is 0.9167 and 0.9014 by using the SDAE with dataset B and dataset A, respectively, which are both higher than the largest value for the SAE (dataset A: 0.6815; dataset B: 0.7424). The value of the Silhouette index in the SDAE is larger than that of the SAE on the whole.

The results of the best cluster number and classification error accuracy at the maximum Silhouette index value by using different models (the testing dataset) are shown in Table 6. In Table 6, the lowest value is 2.22% in the SDAE model with dataset B. For dataset A, it is 4.44% when using

![Figure 14](image-url)
Table 5: The results of the Silhouette index with different cluster numbers by using an SAE/SDAE with adAP (the testing dataset).

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Silhouette index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>SAE</td>
<td>A</td>
<td>0.3302</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.3629</td>
</tr>
<tr>
<td>SDAE</td>
<td>A</td>
<td>0.4920</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.4370</td>
</tr>
</tbody>
</table>

Figure 15: The 3-dimensional clustering results for the testing dataset by using an SAE/SDAE with adAP. (a) SAE-A-testing data-adaptive AP. (b) SAE-B-testing data-adaptive AP. (c) SDAE-A-testing data-adaptive AP. (d) SDAE-B-testing data-adaptive AP.
the SDAE. Therefore, these results prove that the robustness and feature extraction ability of the SDAE suppress the SAE. In addition, adAP can find the available cluster number by itself automatically. The 3-dimensional clustering results for training datasets A and B when using the SAE/SDAE with FCM/GK/GG are shown in Figures 16 and 17. Compared with the SAE, most of the samples are separated well and close to the center point in the SDAE. While some samples

### Table 6: The results of the best cluster number and classification error accuracy at the maximum Silhouette index value by using different models (the testing dataset).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Best clusters no.</th>
<th>Silhouette index</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>SAE-adAP</td>
<td>10</td>
<td>0.6815</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>SDAE-adAP</td>
<td>9</td>
<td>0.9014</td>
<td>4.44</td>
</tr>
<tr>
<td></td>
<td>SAE-adAP</td>
<td>9</td>
<td>0.7425</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>SDAE-adAP</td>
<td>9</td>
<td>0.9167</td>
<td>2.78</td>
</tr>
</tbody>
</table>

Figure 16: The 3-dimensional clustering results for the testing dataset A by using an SAE/SDAE with FCM/GK/GG. (a) SAE-FCM-A. (b) SAE-GK-A. (c) SAE-GG-A. (d) SDAE-FCM-A. (e) SDAE-GK-A. (f) SDAE-GG-A.
exhibit an overlap phenomenon, especially IRF1 and BF1 in Figure 11(a), these samples are separated well by using the SDAE. The classification accuracy by using different combination models for training datasets A and B is shown in Table 7. The lowest error rate is 2.78% with dataset B in the SDAE, which is the same as other combination models, including SAE/SDAE-FCM/GK/GG and SAE-adAP. SAE-adAP is little higher than SAE-GG with dataset B, but adAP can find the available clustering center point automatically.

5. Conclusion

A method based on an SDAE and adAP for bearing fault diagnosis was presented in this study. To reduce the dependence on manual experiments to label data, we used an SDAE without an output layer to extract useful fault features from the frequency domain directly by using FFT decomposition. Additionally, to find the available parameters in

Table 7: The error classification accuracy through various models (the testing dataset).

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FCM</td>
</tr>
<tr>
<td>SAE</td>
<td>A</td>
<td>3.89</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>1.67</td>
</tr>
<tr>
<td>SDAE</td>
<td>A</td>
<td>4.44</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>2.78</td>
</tr>
</tbody>
</table>

Figure 17: The 3-dimensional clustering for the testing dataset B by using an SAE/SDAE with FCM/GK/GG. (a) SAE-FCM-B. (b) SAE-GK-B. (c) SAE-GG-B. (d) SDAE-FCM-B. (e) SDAE-GK-B. (f) SDAE-GG-B.
AP, we introduced adAP for bearing fault diagnosis in this paper. The results show that the performance of the proposed model suppresses other models, such as SAE/SDAE-FCM/GK/GG and SAE-adAP.

The advantages and disadvantages of this paper are as follows.

(1) Using the model proposed in this article can serve to mark different bearing fault signals. For example, the clustering result is used to label the different fault signals, and then an SAE with an output layer can be used to realize online automatic fault diagnosis.

(2) However, the data collected in the actual project contain noise, resulting in the misclassification and mislabeling of the clustering results. Therefore, the classification effect for the subsequent use of an SAE with an output layer is even worse. In order to solve this problem, for future research, we propose an improved SAE model, for example, by adding a data-smoothing model at each hidden layer to eliminate noise layer by layer on the signal, thus improving the accuracy of clustering and classification.

Abbreviations

WT: wavelet transformation
EMD: empirical mode decomposition
IMFs: intrinsic mode functions
EEMD: ensemble empirical mode decomposition
SAE: stacked autoencoder
SDAE: stacked denoising autoencoder
FCM: Fuzzy C-mean
GG: Gustafson–Kessel
ORF: outer race fault
IRF: inner race fault
CC: cluster center
AE: autoencoder
FFT: Fast Fourier transformation.

Data Availability

Previously reported bearing data were used to support this study and are available at (data link: http://csegroups.case.edu/bearingdatacenter/pages/download-data-file). These prior studies (and datasets) are cited at relevant places within the text as references [29].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References


Case Western Reserve University, Bearing Data Center Test Seeded Fault Test Data, Case Western Reserve University, Cleveland, OH, USA, 2013, http://csegroups.case.edu/bearingdatacenter/pages/download-data-file.


