Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Finite-sensor fault-diagnosis simulation study of gas turbine engine using information entropy and deep belief networks

De-long FENG[†‡1], Ming-qing XIAO[1], Ying-xi LIU[2], Hai-fang SONG[1], Zhao YANG[1], Ze-wen HU[1]

([1]*Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China*)

([2]*Air Force Xi'an Flight Academy, Xi'an 710306, China*)

[†]E-mail: fengdelong101@foxmail.com

Received July 5, 2016; Revision accepted Oct. 9, 2016; Crosschecked Nov. 8, 2016

Abstract: Precise fault diagnosis is an important part of prognostics and health management. It can avoid accidents, extend the service life of the machine, and also reduce maintenance costs. For gas turbine engine fault diagnosis, we cannot install too many sensors in the engine because the operating environment of the engine is harsh and the sensors will not work in high temperature, at high rotation speed, or under high pressure. Thus, there is not enough sensory data from the working engine to diagnose potential failures using existing approaches. In this paper, we consider the problem of engine fault diagnosis using finite sensory data under complicated circumstances, and propose deep belief networks based on information entropy, IE-DBNs, for engine fault diagnosis. We first introduce several information entropies and propose joint complexity entropy based on single signal entropy. Second, the deep belief networks (DBNs) is analyzed and a logistic regression layer is added to the output of the DBNs. Then, information entropy is used in fault diagnosis and as the input for the DBNs. Comparison between the proposed IE-DBNs method and state-of-the-art machine learning approaches shows that the IE-DBNs method achieves higher accuracy.

Key words: Deep belief networks (DBNs), Fault diagnosis, Information entropy, Engine
http://dx.doi.org/10.1631/FITEE.1601365 **CLC number:** TP391; V267.3

1 Introduction

A gas turbine engine is the heart of an aircraft. Frequently, passenger planes experience flight delays and military aircrafts have a forced grounding due to engine failure. Engine fault diagnosis is attracting increasing attention because of rising maintenance costs and the importance of flight safety. A gas turbine engine is a complicated thermal rotating machinery system which works in a harsh environment of high temperature, high pressure, and high rotating speed. Thus, the maintenance of a gas turbine engine involves characteristics of multiple failure modes, a large number of engine components, multi-mode failure in composites, ensuring a long operation life for the engine. Engine fault diagnosis is a basic part of engine maintenance. The main focus of this paper is to enhance the accuracy rate and test speed of fault diagnosis using a finite number of sensors.

The state parameters of the testing system often contain important features that reflect the system operating state. Generally, some signal-processing based approaches are used in fault diagnosis to obtain the characteristic information of the measured object (Pan *et al*., 2015). For example, we can obtain the statistical parameters of the signal and the gist of spectral characteristics for system status analysis and diagnosis by time or frequency domain analysis (Aguiar and Guedes, 2015; Rastegin, 2015; Song *et al*., 2015). The nature of the signal analysis process is to reflect the multi-level interior features of a signal by the use of different conversion methods in a different transform domain (Geng *et al*., 2006; Su and You, 2014; Sekerka, 2015). Ferrer (2007) discussed

‡ Corresponding author
ORCID: De-long FENG, http://orcid.org/0000-0002-6274-0720

the problem of data that often exhibits high correlation, rank deficiency, low signal-to-noise ratio, and missing values, and advocated the use of multivariate statistical process control based on principal component analysis (MSPC-PCA) as an efficient statistical tool for process understanding, monitoring and diagnosing assignable causes for special events. Xie and Zhang (2005) proposed a fault diagnosis approach using support vector machine (SVM) with data dimension reduction by PCA and linear discriminant analysis (LDA) methods, where PCA and LDA are separately applied to reduce the data dimensions and extra data features from the raw data. We can directly derive features of the system status using the above time domain or frequency domain analysis methods when the form of a signal is simple and the interrelated features of the signal are clear, such as rotating machinery vibration signals for a stationary cycle (Dai and Tian, 2013; Jin *et al*., 2014). However, it is difficult to extract a feature directly using the normal signal conversion method when the form of a signal is complex, such as rotating machinery vibration signals for a nonlinear or abnormal non-steady state due to changing operating parameters. Therefore, further investigation into signal analysis methods is required to achieve automatic extraction of features and quantitative characterization of the corresponding parameters for complex signals.

For machine learning, original data feature extraction is significant and is the most time-consuming. Supervised learning is a kind of learning frame including prior tasks where computers obtain models through labeled data. As a new branch of computer learning, deep learning learns the features from a data skipping feature in the design stage (Bengio, 2009). Most methods of deep learning are based on neural networks. For this kind of deep learning, complicated high-level constructions depend on superposing many nonlinear neural models (Bengio *et al*., 2013).

Deep learning in the field of machine learning approaches artificial intelligence (AI) to an extraordinary degree. The motivation for deep learning, which belongs to supervised learning, is to build and simulate the neural network of the human brain for learning analysis (Rodríguez *et al*., 2013). The concept of deep learning is derived from research into artificial neural networks. Deep learning, which, relative to simple learning, includes most classification and regressive algorithms, is restricted to the descriptive ability of complex functions in the case of finite samples and computing units (Saimurugan *et al*., 2011; Sainath *et al*., 2013; Zhou *et al*., 2014). Deep learning can approximate complex functions to describe input data regularities for distribution and to study essential characteristics of a dataset from a minor portion by focusing on a sample and learning a kind of deep nonlinear network structure (Ong *et al*., 2014).

To represent the data distribution feature, deep learning simulates more nervous layer activities that abstract assembling low- to high-level attributive features (Larochelle *et al*., 2009). Deep learning was proposed by Hinton *et al*. (2006). To solve the problem of structure optimization based on deep belief networks (DBNs), unsupervised greedy layer-by-layer learning was proposed. Hinton *et al*. (2012) further proposed a deep structure, called a 'multi-layer automatic encoder'. It is a feed-forward neural network that can predict the input of itself. In addition, Sermanet *et al*. (2012) proposed convolution neural networks (CNNs), being the first truly multi-layer structural learning algorithm. CNNs improve training performance through decreasing the number of parameters by spatially relative relationships. CNNs are different from DBNs in that CNNs belong to discriminative training algorithms. CNNs are created based on the requirement of minimizing the pretreatment data for the deep learning framework (Zhang *et al*., 2015). Due to the influence of an early time delayed neural network, CNNs reduce the complexity by sharing the weight of the time domain (Sainath *et al*., 2015). Altogether, CNNs have achieved good performance in several experiments.

There are many training methods for deep neural networks. The traditional training method is stochastic gradient descent (SGD) (Bottou, 2012). This method adds only one training sample into one training process. The loss function will be reduced through the training sample. The back propagation algorithm can be used to calculate the value of the gradient of the loss function (Niu *et al*., 2014). There are many other convex optimization approaches that can be guaranteed to find the global minimum. However, if the deep neural network were a non-convex optimization problem, the curvature of

the loss function would tend to the extremum. In fact, the Newton method is very suitable for dealing with the curvature problem, but it cannot train a large-scale problem such as a neural network. There are two reasons: one is that the Newton method has to use all the samples for each training; the other is that it needs to build a Hessian matrix and to find the inverse matrix, which leads to a huge amount of computation. Thus, for a non-convex optimization problem, we usually refer to the solution ideas from convex optimization (Bengio, 2012; Martens and Sutskever, 2012).

Based on the discussion above, in this paper we propose a method for deep belief networks based on information entropy, IE-DBNs, to diagnose engine faults. The method presents the complexity of the information entropy of the generalized transformation space as the input for the deep belief network. The entropy is combined with the time domain, frequency domain, and time-frequency domain signal conversion methods, respectively. The single- and multi-signal information entropy is established as a feature extraction method in the different transformation spaces. To effectively describe the signal feature and quantitatively calculate the intrinsic signal characteristics at different levels, especially for the nonlinear and non-stationary signal, we discuss a variety of complexity information entropy indicators, such as singular spectrum entropy, power spectrum entropy, and multi-resolution singular spectrum entropy. We also provide quantitative indicators and evaluation based on fault diagnosis and condition monitoring. For finite sensory signals, a logistic regression layer is added on top of the deep belief networks for fault classification.

2 Information entropy features

Information entropy is a quantitative evaluation for the uncertainty of a system state, and it has a strong ability to describe the internal system information. Researchers have tried to use information entropy to extract the features of the operating status of a system in the field of mechanical fault diagnosis (Nichols *et al*., 2006; Susan and Hanmandlu, 2013; Li *et al*., 2016).

2.1 Singular spectrum entropy (time domain)

A singular spectrum is a modern spectral analysis technique based on kinetic analysis. Its basic idea is to reconstruct the phase space and decompose the singular value of a time-domain signal sequence for a system, and then to obtain the complexity features to describe the system state. The basic principle is as follows (Cui *et al*., 2009): For a discrete time signal, the time delay embedding method is used to reconstruct the phase space. The signal $\boldsymbol{X}_i$ (i=1, 2, …, I) is mapped to the phase space where the length is m and the delay constant of the analysis window is τ. Thus, the track matrix is defined as

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{X}_1\\ \boldsymbol{X}_2\\ \vdots\\ \boldsymbol{X}_I\end{bmatrix}=\begin{bmatrix}x_1 & x_2 & \cdots & x_m\\ x_{\tau+1} & x_{\tau+2} & \cdots & x_{\tau+m}\\ \vdots & \vdots & & \vdots\\ x_{(I-1)\tau+1} & x_{(I-1)\tau+2} & \cdots & x_{(I-1)\tau+m}\end{bmatrix}. \quad (1)$$

According to the singular value decomposition principle, there must be an $m\times l$-dimensional matrix $\boldsymbol{U}$, an $l\times l$-dimensional diagonal matrix $\boldsymbol{\Lambda}$, and an $l\times n$-dimensional matrix $\boldsymbol{V}$ for an $m\times n$-dimensional real matrix $\boldsymbol{A}$:

$$\boldsymbol{A}_{m\times n}=\boldsymbol{U}_{m\times l}\boldsymbol{\Lambda}_{l\times l}\boldsymbol{V}^{\mathrm{T}}_{n\times l}, \quad (2)$$

where the main diagonal element λ_i of $\boldsymbol{\Lambda}$ is the matrix $\boldsymbol{A}$'s singular value, which is non-negative and $\lambda_1\geqslant\lambda_2\geqslant\ldots\geqslant\lambda_l\geqslant 0$. The matrix $\boldsymbol{\Lambda}$ can be described as

$$\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\ldots,\lambda_k,0,\ldots,0). \quad (3)$$

When the signal is interfered with the ambient noise or has a low signal-to-noise ratio, all the diagonal elements of matrix $\boldsymbol{\Lambda}$ may be nonzero values. The more nonzero elements in the main diagonal, the more complexity the signal components will have. Furthermore, the number k of nonzero singular values in matrix $\boldsymbol{\Lambda}$ reflects the number of different modes contained by the track matrix, and the value of λ_i reflects the proportion of the total mode. Therefore, we can obtain the singular spectrum entropy H_s through the λ_i.

$$H_\mathrm{s}=-\sum_{j=1}^{m}p_j\ln p_j,\ \ p_j=\lambda_j\Big/\sum_{j=1}^{L}\lambda_j. \quad (4)$$

Singular spectrum entropy H_s means the uncertainty level of each mode that is divided by a time-domain signal sequence using a singular spectrum, and it also means the complexity of the signal energy distribution in the time domain.

2.2 Power spectrum entropy (frequency domain)

The most common spectral analysis methods are amplitude spectrum, phase spectrum, power spectrum, and cepstrum analysis (Li, 2015). Power spectrum analysis in the signal transform space is a feature extraction method based on the frequency domain and information entropy. For the discrete signal sequence x_n (n=0, 1, …, N−1), the power spectrum is defined as

$$\hat{S}(\omega)=\frac{1}{N}\left|\sum_{n=0}^{N-1}x_n\mathrm{e}^{-\mathrm{j}\omega n}\right|^2. \tag{5}$$

It can also be written as

$$\hat{S}(\omega)=\frac{1}{N}\left|X(\omega)\right|^2, \tag{6}$$

where $X(\omega)$ is the Fourier transform of sequence $\{x_n\}$. It is a process of transforming a signal from the time domain to the frequency domain according to the Parseval theorem, described by

$$\sum_{n=0}^{N-1}\left|x_n\right|^2=\sum_{k=0}^{N-1}\left|S(k)\right|^2, \tag{7}$$

where S_k (k=0, 1, …, N−1) is an energy distribution of the original signal in the frequency domain. Thus, the power spectrum entropy can be defined as

$$H_{\mathrm{p}}=-\sum_{k=0}^{K-1}p_k\ln p_k,\quad p_k=S_k\Big/\sum_{k=1}^{N}S_k. \tag{8}$$

The power spectrum entropy reflects the complexity of the energy distribution of the signal in the frequency domain.

2.3 Wavelet energy spectrum entropy (time-frequency domain)

The wavelet transform inherits the method of locating the window Fourier transform, and it has the ability to analyze the varying locations of time and frequency in different scales, which is called a multi-resolution characteristic (Koverda and Skokov, 2012; Liu *et al*., 2014; Zhao and Ye, 2016).

For the finite energy signal $f(t)$, the wavelet transform is expressed as follows:

$$W_f(a,b)=<f,\psi_{a,b}>=\left|a\right|^{-\frac{1}{2}}\int_{-\infty}^{+\infty}f(t)\psi\left(\frac{t-b}{a}\right)\mathrm{d}t, \tag{9}$$

where the basis function $\psi_{a,b}(t)$ can be expressed as

$$\psi_{a,b}(t)=\left|a\right|^{-\frac{1}{2}}\varphi\left(\frac{t-b}{a}\right),\quad a,b\in\mathbb{R},\ a\neq 0, \tag{10}$$

where a and b are the scale parameter and translation parameter respectively, and $\Psi(t)\in L^2(\ |)$ is called a wavelet function satisfying the admissible condition:

$$C_\psi=\int_{-\infty}^{+\infty}\left|\Psi(\omega)\right|^2\left|\omega\right|^{-1}\mathrm{d}\omega<+\infty, \tag{11}$$

where $\Psi(\omega)$ is the Fourier transform of $\Psi(t)$.

The scale parameter a and translation parameter b of the wavelet function can be adjusted according to the shape and size of the wavelet window. We may change the center frequency of the window, the width of the sub-wavelet, and the position in the time domain. This allows the wavelets to have both an ability of locating the spatial domain and the frequency domain, and a positive 'zoom' feature. The wavelet function effectively reflects the local mutation information of non-stationary signals.

For the finite energy signal $f(t)$, the energy conservation of the wavelet transform is

$$\int_{-\infty}^{+\infty}\left|f(t)\right|^2\mathrm{d}t=\frac{1}{C_\psi}\int_0^\infty a^{-2}E(a)\mathrm{d}a, \tag{12}$$

where $E(a)$ is the energy value of $f(t)$ in the a scale, called the wavelet energy spectrum:

$$E(a)=\int_{-\infty}^{+\infty}\left|W_f(a,b)\right|^2\mathrm{d}b. \tag{13}$$

Thus, the signal $f(t)$ is decomposed into the wavelet energy spectrum $\boldsymbol{E}$=[E_1, E_2, …, E_n] in the n scale which is a division of the energy in the time-frequency domain. The wavelet energy spectrum

entropy H_w is defined as

$$H_w=-\sum_{i=1}^{n}p_i\ln p_i,\quad p_i=E_i\Big/\sum_{i=1}^{n}E_i. \tag{14}$$

2.4 Multi-resolution entropy

Based on multi-resolution ideas and the Mallat orthogonal wavelet decomposition algorithm, the calculation methods for discrete wavelet series of a signal can be briefly stated as follows. For the arbitrary signal $f(t)\in L^2(\ )$, the mark is introduced:

$$\begin{cases}c_{j,k}=\int_{\mathbb{R}}f(t)\overline{\varphi}_{j,k}(t)\mathrm{d}t,\\ d_{j,k}=\int_{\mathbb{R}}f(t)\overline{\psi}_{j,k}(t)\mathrm{d}t,\end{cases} \tag{15}$$

where $c_{j,k}$ is the scaling coefficient of $f(t)$, and $d_{j,k}$ is the wavelet coefficient of $f(t)$. For arbitrary integers j and k, the scaling function $\varphi(t)$ and wavelet function $\psi(t)$ are

$$\begin{cases}\varphi_{j,k}(t)=2^{j/2}\varphi(2^jt-k),\\ \psi_{j,k}(t)=2^{j/2}\psi(2^jt-k).\end{cases} \tag{16}$$

The orthogonal projections of $f(t)$ in the closed subspaces V_j and W_j are denoted by $A_jf(t)$ and $D_jf(t)$, respectively:

$$\begin{cases}A_jf(t)=\sum_{k\in\mathbb{Z}}c_{j,k}\varphi_{j,k}(t),\\ D_jf(t)=\sum_{k\in\mathbb{Z}}d_{j,k}\psi_{j,k}(t).\end{cases} \tag{17}$$

According to the orthogonal direct sum decomposition $V_{j+1}=V_j\oplus W_j$, we obtain

$$A_{j+1}f(t)=A_jf(t)+D_jf(t), \tag{18}$$

where $A_jf(t)$ is the low-frequency approximation of $f(t)$ in scale 2^{-j} and $D_jf(t)$ is the high-frequency component of $f(t)$ in scale 2^{-j}. Suppose the discrete wavelet coefficient of multi-resolution entropy is $D=\{d(k): k=1, 2, \ldots, N\}$ in scale j, and a slide window W is defined based on the wavelet coefficient. Suppose the width of the window is $w\in\ $, and the slide factor is $\delta\in\ $. Then we have

$$W(m;w,\delta)=\{d(k):k=1+m\delta,2+m\delta,\ldots,w+m\delta\}, \tag{19}$$

where m is the number of windows moved. The slide window is divided into L intervals:

$$W(m;w,\delta)=\bigcup_{l=1}^{L}Z_l, \tag{20}$$

where $\{Z_l=(S_{l-1}, S_l): l=1, 2, \ldots, L\}$ is mutually disjoint and $S_0<S_1<S_2<\ldots<S_L$.

$$\begin{cases}S_0=\min[W(m;w,\delta)]\\ \quad=\min[\{d(k):k=1+m\delta,2+m\delta,\ldots,w+m\delta\}],\\ S_L=\max[W(m;w,\delta)]\\ \quad=\max[\{d(k):k=1+m\delta,2+m\delta,\ldots,w+m\delta\}].\end{cases} \tag{21}$$

The multi-resolution entropy is defined as

$$H_r=-\sum_{l=1}^{L}p^m(Z_l)\ln p^m(Z_l),\quad m=1,2,\ldots,M, \tag{22}$$

where $p^m(Z_l)$ is the probability of $d(k)$ falling in the Z_l interval. The multi-resolution entropy has unique sensitivity and location detection capability for small parameters in the dynamics system.

2.5 Multi-resolution singular spectrum entropy

This subsection presents a multi-resolution singular spectrum entropy model, which has the following functions and characteristics (Nourani *et al.*, 2015): (1) The energy distribution of singular characteristics can get into any local band based on the calculation of the wavelet coefficients of singular spectrum entropy in different frequency bands through wavelet packet decomposition of the signal; (2) This method has better nonlinear feature extraction and noise suppression capabilities; (3) Through the selection and optimization of parameters, this method can improve the analytical performance and the computing efficiency of the measurement model.

First, the signal should be decomposed by the wavelet. The Mallat algorithm of orthogonal wavelet transform is used for wavelet packet decomposition.

For an arbitrary signal $f(t)\in L^2(\ )$, we have

$$d_{j,k}^{l}=\int_{\mathbb{R}} f(t)\overline{u}_{l;j,k}(t)\mathrm{d}t. \tag{23}$$

The projection of $f(t)$ in the wavelet space can be described as

$$D_j^m f(t)=\sum_{k\in\mathbb{Z}} d_{j,k}^m u_{m;j,k}(t). \tag{24}$$

The formula of the Mallat algorithm for wavelet packet decomposition is

$$\begin{cases} d_{j,k}^{2m}=\sum_{l\in\mathbb{Z}}\overline{h}_{l-2k}d_{j+1,l}^m, \\ d_{j,k}^{2m+1}=\sum_{l\in\mathbb{Z}}\overline{g}_{l-2k}d_{j+1,l}^m. \end{cases} \tag{25}$$

The frequency band of signal $f(t)$ is unceasingly subdivided through the decomposition of scales 1, 2, …, J. Additionally, if $f(t)$ is decomposed by j layers of the wavelet packet, the kth node coefficient of the jth layer is

$$D_{j,k}=\left\{d_{j,k}(m):m=1,2,\ldots,2^{-j}N\right\}. \tag{26}$$

Then the singular spectrum value of every node coefficient of every scale can be calculated, and the time-delay embedding method is used to reconstruct the matrix $\boldsymbol{A}_{j,k}$ (see Eq. (27)).

Therefore, the multi-resolution singular spectrum entropy is

$$H_{j,k}(f)=-\sum_{m=1}^{m_0} p_{j,k}^m \ln p_{j,k}^m,\ p_{j,k}^m=\lambda_{j,k}^m\Big/\sum_{m=1}^{m_0}\lambda_{j,k}^m. \tag{28}$$

If the range of the frequency distribution of signal $f(t)$ is $[\omega_\mathrm{d}, \omega_\mathrm{g}]$, $H_{j,k}(f)$ will reflect the energy distribution of the singularity of signal $f(t)$ in the band of $[\omega_\mathrm{d}+(k-1)(\omega_\mathrm{g}-\omega_\mathrm{d})/2^j, W_\mathrm{d}+k(\omega_\mathrm{g}-\omega_\mathrm{d})/2^j]$ according to the principle of wavelet packet decomposition.

3 Fault diagnosis using information entropy and deep belief networks

Currently, most classification and regression learning methods have shallow structures. Their limitation is a deficient capacity for modeling complex functions under a limited number of samples and calculating units, and the generalization ability is restricted. For deep learning, the idea is to stack a plurality of layers to describe the complex functions (Hinton, 2010). That is to say, the output of one layer becomes the input of the next layer. In this way, we can achieve a classification expression for complex input information. DBNs, one of the deep learning methods, can solve the problems of training a neural multi-layer network, which is difficult for traditional back propagation (BP) algorithms (Memisevic and Hinton, 2010): (1) From top to bottom, the gradient becomes more sparse and the error correction signal becomes smaller; (2) The weights easily converge to local minima; (3) The labeled data is trained in general.

3.1 Restricted Boltzmann machines

Suppose there is a bipartite graph (Fig. 1) in which there are no links between the nodes in each layer. One is a visible layer called the input data layer ($\boldsymbol{v}$), and the other is a hidden layer ($\boldsymbol{h}$). It is assumed that all the nodes are random binary variable nodes (taking only the value of 0 or 1), and suppose the full probability distribution $p(\boldsymbol{v}, \boldsymbol{h})$ is a Boltzmann distribution. We can call this model an RBM, which is a typical neural network where the units of the visible layer and the hidden layer are interconnected, and the hidden units can achieve high-order dependence on the visible units (Sutskever *et al*., 2008). Compared with a conventional sigmoid network, the weights of an RBM are easy to train.

Let us discuss why RBM is called a deep learning method. First, as a bipartite graph, all the hidden nodes are conditionally independent when $\boldsymbol{v}$ is known,

$$\boldsymbol{A}_{j,k}=\begin{bmatrix} d_{j,k}(1) & d_{j,k}(2) & \cdots & d_{j,k}(M) \\ d_{j,k}(2) & d_{j,k}(3) & \cdots & d_{j,k}(M+1) \\ \vdots & \vdots & & \vdots \\ d_{j,k}(2^{-j}N-M+1) & d_{j,k}(2^{-j}N-M+2) & \cdots & d_{j,k}(2^{-j}N) \end{bmatrix}. \tag{27}$$

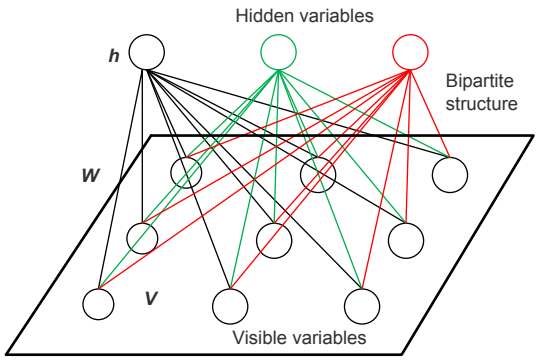

Fig. 1 The structure of a restricted Boltzmann machine

namely $P(\boldsymbol{v}|\boldsymbol{h})=p(\boldsymbol{h}_1|\boldsymbol{v})p(\boldsymbol{h}_2|\boldsymbol{v})\ldots p(\boldsymbol{h}_n|\boldsymbol{v})$. Similarly, all the visible nodes are conditionally independent when the hidden layer $\boldsymbol{h}$ is known. At the same time, both $\boldsymbol{v}$ and $\boldsymbol{h}$ satisfy the Boltzmann distribution. Therefore, the hidden layer $\boldsymbol{h}$ can be obtained by $p(\boldsymbol{h}|\boldsymbol{v})$ when the input is $\boldsymbol{v}$, and then the visible layer can be obtained by $p(\boldsymbol{h}|\boldsymbol{v})$ (Fig. 2). If the original visible layer $\boldsymbol{v}$ and the visible layer $\boldsymbol{v}_1$ obtained from the hidden layer are the same according to some adjustment of the parameters, the hidden layer obtained will be a different expression of the visible layer. Thus, the hidden layer can be used as the feature of the input data of the visible layer, and it is a kind of deep learning method.

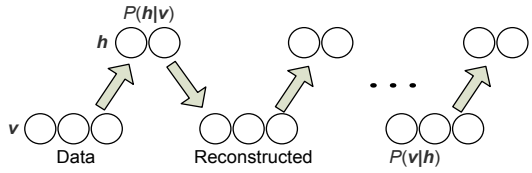

Fig. 2 The principle of a restricted Boltzmann machine

Now we need to do some mathematical analysis to determine the weights between the visible layer node and the hidden layer node. The energy of a joint configuration can be expressed as (Mohamed *et al.*, 2012)

$$E(\boldsymbol{v},\boldsymbol{h};\theta)=-\sum_{ij}W_{ij}v_ih_j-\sum_i b_iv_i-\sum_j a_jh_j, \quad (29)$$

where $\theta=\{\boldsymbol{W}, a, b\}$ are the model parameters.

The joint probability distribution of the configuration can be determined by Boltzmann distribution and the energy of this configuration:

$$\begin{aligned}P_\theta(\boldsymbol{v},\boldsymbol{h})&=\frac{1}{Z(\theta)}\exp(-E(\boldsymbol{v},\boldsymbol{h};\theta))\\&=\frac{1}{Z(\theta)}\prod_{ij}\mathrm{e}^{W_{ij}v_ih_j}\prod_i\mathrm{e}^{b_iv_i}\prod_j\mathrm{e}^{a_jh_j},\end{aligned} \quad (30)$$

where $1/Z(\theta)$ is the partition function and $\mathrm{e}^{W_{ij}v_ih_j}$ is the potential function.

$$Z(\theta)=\sum_{\boldsymbol{h},\boldsymbol{v}}\exp(-E(\boldsymbol{v},\boldsymbol{h};\theta)). \quad (31)$$

The hidden nodes are conditionally independent:

$$P(\boldsymbol{h}|\boldsymbol{v})=\prod_j P(h_j|\boldsymbol{v}). \quad (32)$$

Then the probability of 1 or 0 of the jth hidden layer can be easily obtained based on the known visible layer $\boldsymbol{v}$ by factorizing Eq. (32):

$$P(h_j=1|\boldsymbol{v})=\left[1+\exp\left(-\sum_i W_{ij}v_i-a_j\right)\right]^{-1}. \quad (33)$$

Similarly, based on the known hidden layer $\boldsymbol{h}$, we have

$$P(\boldsymbol{v}|\boldsymbol{h})=\prod_i P(v_i|\boldsymbol{h}), \quad (34)$$

$$P(v_i=1|\boldsymbol{h})=\left[1+\exp\left(-\sum_j W_{ij}h_j-b_i\right)\right]^{-1}. \quad (35)$$

Given a sample set $D=\{v(1), v(2), \ldots, v(N)\}$ that satisfies the independent distribution, the parameters $\theta=\{\boldsymbol{W}, a, b\}$ need to be calculated. Maximize the following log-likelihood function:

$$L(\theta)=\frac{1}{N}\sum_{n=1}^{N}\log P_\theta(\boldsymbol{v}^{(n)})-\frac{\lambda}{N}\|\boldsymbol{W}\|_{\mathrm{F}}^2. \quad (36)$$

The parameter $\boldsymbol{W}$ can be obtained by the derivation of the maximum log-likelihood function when L is the maximum.

$$\frac{\partial L(\theta)}{\partial W_{ij}}=E_{P_{\mathrm{data}}}[v_ih_j]-E_{P_\theta}[v_ih_j]-\frac{2\lambda}{N}W_{ij}. \quad (37)$$

If we increase the number of hidden layers, we can obtain a deep Boltzmann machine (DBM). If we use a Bayesian belief network near the visible layer (i.e., a directed graph model), and at the same time a restricted Boltzmann machine is used far from the visible layer, we can obtain DBN. Fig. 3 illustrates the differences between DBN and DBM.

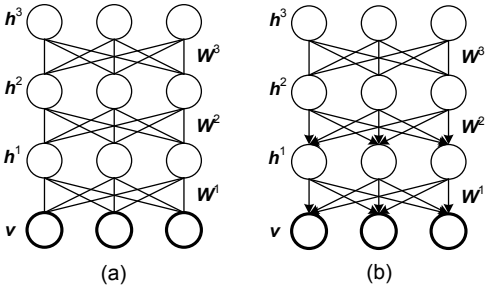

Fig. 3 The difference between the deep Boltzmann machine (a) and deep belief network (b)

3.2 Deep belief networks

DBNs are composed of many restricted Boltzmann machines, which are like the building blocks as shown in Fig. 4. The connections of a DBN are determined via the top-down generated weights. In the beginning or pre-training step, the weight of the generation model can be obtained through an unsupervised greedy layer-by-layer method, and this method has been proved effective by Hinton, who called this method 'contrastive divergence' (Tran *et al*., 2014; Chen *et al*., 2015). During the training phase, the hidden units are trained to obtain high-order dependence of the visible units, and a vector $\boldsymbol{v}$ is generated in the visible layer and passed to the hidden layer (Tamilselvan *et al*., 2011). In return, the input of the visible layer is randomly selected to reconstruct the original input signal. Finally, this new visible neuron activation unit passes forward and reconstructs the hidden layer activation unit to obtain $\boldsymbol{h}$. In the training process, however, the first step is that the visible vector maps to the hidden units, and then the visible units are reconstructed by the hidden layer units. The new hidden units can obtain these new visible units and map to the hidden units again. The implementation of such repeated steps is called Gibbs sampling. The correlation difference between the hidden layer activation unit and the input of the visible layer is the main basis of weight updates (Tamilselvan and Wang, 2013).

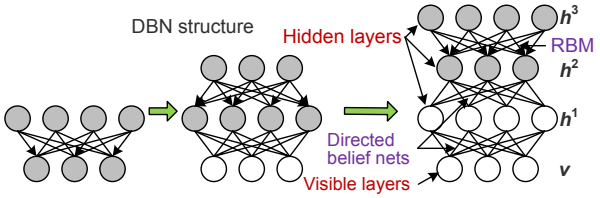

Fig. 4 Restricted Boltzmann machines constitute the deep belief networks

The added layer of the network will increase the logarithm probability of the training data. In other words, more layers in the network will give a more accurate expression of the energy. In addition, it will reduce the training time, because a single step can achieve the learning of the maximum likelihood.

Fig. 5 shows the framework for DBNs. In the top two layers, the weights are connected together. The output of the lower layers can provide a reference clue or is related to the top layer, and then the top layer will be linked to the memory content. After pre-training, DBNs can use labeled data and BP algorithms to fine-tune the discrimination result. A label set will be attached to the top (the promotion of associative memory), and the classification facets of the network can be obtained by the recognized weights learned through a top-down approach. The performance obtained from this approach is better than that from a simple BP algorithm. The reason is that only one local search of the parameter space is required. Compared with feed-forward neural networks, the DBNs method exhibits better performance in terms of training time and convergence time.

3.3 Training method for the deep belief networks model

The deep belief networks training process contains layer-by-layer supervised learning, back-propagation learning, and fine-tuning.

First, we use unsupervised learning starting from the bottom layer moving to the top layer, that is, a layer-by-layer training method. The training data is used to train the parameters of each layer of the DBNs, which can be seen as an unsupervised training process. The largest difference between DBNs and traditional neural networks is that DBNs can be seen as a feature learning process.

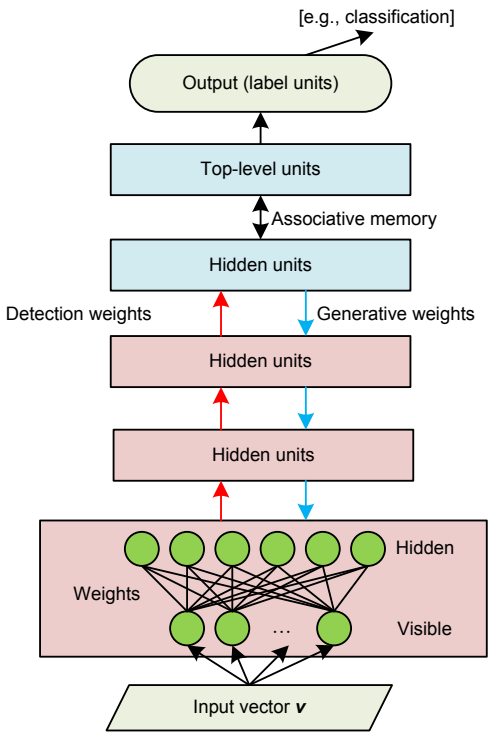

Fig. 5 The framework of deep belief networks

The process can be described in detail as follows. At the beginning, the training data is used to train the parameters of the first layer. Next, we will obtain the hidden layer of a three-layer neural network, which has a minimum difference between the output and the input. The model obtained can learn the structure of the data by itself because of the restricted capacity and sparsity of the model, and thus we can obtain the features that have stronger expression ability than the input. After learning and obtaining the parameters of the (n−1)th layer, the output of the (n−1)th layer will be the input of the nth layer, and then the training of the nth layer is conducted.

In the next part, back propagation learning is used to fine-tune the trained DBNs. It is a top-down supervision and learning method where the labeled data is used to train and fine-tune the network. In addition, errors are transmitted from the top layer to the bottom layer.

The first step of the training method is similar to the random initialization process of neural networks. However, the initial value of deep learning is that it learns the structure of the input data, which is closer to the global optimum. Thus, the high accuracy of deep learning is largely due to feature learning in the first step.

3.4 Deep belief networks based on information entropy

Based on the discussion above, we propose a gas turbine engine fault diagnosis method which uses information entropy and deep belief networks. In this method, the information entropy is not the single entropy introduced in Section 2. The information entropy of IE-DBNs, which is called joint complexity information entropy, is a multivariate feature extraction method based on the complexity of information entropy. The analytical and diagnostic processes of the system operating state rely not only on the inherent characteristics of a single signal, but also on the relevant characteristics of the parameters of multi-source signals or the coupling state of the related signal, especially for fault diagnosis for nonlinear complex systems. The information entropy is extended to a multi-dimensional space, and the probability of component distribution of the energy distribution of a single variable is expanded to a joint probability distribution of multivariable energy. This is then the model for joint complexity information entropy, which achieves multivariable feature extraction based on a broad signal space and can evaluate the signal associated properties of energy distribution.

The joint energy distribution and component distribution probability of a two-dimensional random signal is defined as follows. It has a generalized signal sequence $F_x=\{f(x_i^m)\colon i=1, 2, \ldots, N; m=1, 2, \ldots, M\}$ and $G_y=\{g(y_j^m)\colon j=1, 2, \ldots, N; m=1, 2, \ldots, M\}$. Each of these is present in the same kind of signal spaces T_x and T_y. If F_x and G_y are divided into R and S feature subspaces T_x^r (r=1, 2, …, R) and T_y^s (s=1, 2, …, S) within T_x and T_y, respectively, they can constitute the joint distribution space $T_{(X,Y)}$, and $T_{(X,Y)}$ can be divided into the $R\times S$ joint feature subspaces which are orthogonal to each other. This is expressed as $T_{(X,Y)}^{rs}=\{T_X^r, T_Y^s\}$ (r=1, 2, …, R; s=1, 2, …, S). Thus, the joint component probability $p_{F_XG_Y}(r,s)$ of each feature subspace $T_{(X,Y)}^{rs}$ is

$$p_{F_XG_Y}(r,s)=\frac{E_{F_XG_Y}(r,s)}{\sum_{r=1}^{R}\sum_{s=1}^{S}E_{F_XG_Y}(r,s)},\quad r=1,2,\ldots,R,\ s=1,2,\ldots,S, \tag{38}$$

where $E_{F_XG_Y}(r,s)$ is the generalized joint energy function of F_X and G_Y. In addition, $\sum_{r=1}^{R}\sum_{s=1}^{S}E_{F_XG_Y}(r,s)\neq 0$.

$$E_{F_x}(r)=\sum_{s=1}^{S}E_{F_XG_Y}(r,s),\ E_{G_Y}(s)=\sum_{r=1}^{R}E_{F_XG_Y}(r,s). \quad (39)$$

Therefore, $p_{F_X}(r)$ and $p_{G_Y}(s)$ are the edge component probabilities of $T_{(X,Y)}^{rs}$ for T_X^r and T_Y^s, respectively:

$$p_{F_X}(r)=\frac{E_{F_X}(r)}{\sum_{r=1}^{R}E_{F_X}(r)},\quad p_{G_Y}(s)=\frac{E_{G_Y}(s)}{\sum_{s=1}^{S}E_{G_Y}(s)}. \quad (40)$$

For the two-dimensional generalized random signal sequence (F_X, G_Y), the joint complexity information entropy $H_j(F_XG_Y)$ and edge complexity information entropy $H_j(F_X)$, $H_j(G_Y)$ are defined as

$$H_j(F_XG_Y)=-\sum_{r=1}^{R}\sum_{s=1}^{S}p_{F_XG_Y}(r,s)\ln p_{F_XG_Y}(r,s), \quad (41)$$

$$H_j(F_X)=-\sum_{r=1}^{R}p_{F_X}(r)\ln p_{F_X}(r), \quad (42)$$

$$H_j(G_Y)=-\sum_{s=1}^{S}p_{G_Y}(s)\ln p_{G_Y}(s), \quad (43)$$

where $H_j(F_XG_Y)$ reflects the feature of joint energy distribution, and the complexity levels of (F_X, G_Y), $H_j(F_X)$ and $H_j(G_Y)$, reflect the energy distribution of F_X and G_Y, respectively. To obtain $H_j(F_XG_Y)$ and $p_{F_XG_Y}(r,s)$, the joint energy function $E_{F_XG_Y}(r,s)$ of each feature subspace needs to be calculated. The $E_{F_XG_Y}(r,s)$ is defined as

$$E_{F_XG_Y}(r,s)=E_r\cdot E_s,\ r=1,2,...,R,\ s=1,2,...,S, \quad (44)$$

where E_r and E_s are nonnegative bounded generalized energy functions of F_x in subspace T_x and of G_y in subspace T_y, respectively. The calculation method depends on the specific form of the generalized signal space and the division manner of the feature space. According to the signal characteristic and the analysis requirement, the calculation method can use the energy function of the information entropy of the different conversion spaces. Therefore, the calculation method for univariate entropy is the basis for the calculation of joint complexity information entropy. For example, there are two signal spectrums $X(f_r)$ (r=1, 2, …, R) and $Y(f_s)$ (s=1, 2, …, S) in the frequency domain, where each frequency of a discrete spectrum of the signal can be regarded as a feature subspace. Furthermore, $|X(f_r)|^2$ and $|Y(f_s)|^2$ represent the energy of X in f_r and of Y in f_s, respectively. Thus, the joint energy function is

$$E_{F_XG_Y}(r,s)=\left|X(f_r)\right|^2\cdot\left|Y(f_s)\right|^2. \quad (45)$$

Thus, we can obtain $p_{F_XG_Y}(r,s)$ and $H_j(F_XG_Y)$. Then, the probability of 1 or 0 of the jth node of the hidden layer of IE-DBNs is

$$\begin{aligned}&P\left(h_j=1\middle|H_j(F_XG_Y)\right)\\&=\frac{1}{1+\exp\left(-\sum_i W_{ij}\left|H_{ji}(F_XG_Y)-a_j\right.\right)}\\&=\frac{1}{1+\exp\left(-\sum_i W_{ij}\left|-\sum_{r=1}^{R}\sum_{s=1}^{S}p_{F_XG_Y}(r,s)\ln p_{F_XG_Y}(r,s)-a_j\right.\right)}.\end{aligned} \quad (46)$$

So, the parameter $\boldsymbol{W}$ of IE-DBNs can be obtained by

$$\begin{aligned}\frac{\partial L(\theta)}{\partial W_{ij}}&=E_{P_{\text{data}}}\left[H_{ji}(F_XG_Y)h_j\right]\\&\quad-E_{P_\theta}\left[H_{ji}(F_XG_Y)h_j\right]-\frac{2\lambda}{N}W_{ij}\\&=E_{P_{\text{data}}}\left\{\left[-\sum_{r=1}^{R}\sum_{s=1}^{S}p_{F_XG_Y}(r,s)\ln p_{F_XG_Y}(r,s)\right]h_j\right\}\\&\quad-E_{P_\theta}\left\{\left[-\sum_{r=1}^{R}\sum_{s=1}^{S}p_{F_XG_Y}(r,s)\ln p_{F_XG_Y}(r,s)\right]h_j\right\}\\&\quad-\frac{2\lambda}{N}W_{ij}.\end{aligned} \quad (47)$$

The framework for IE-DBNs is shown in Fig. 6. A logistic regression layer is added after the top layer

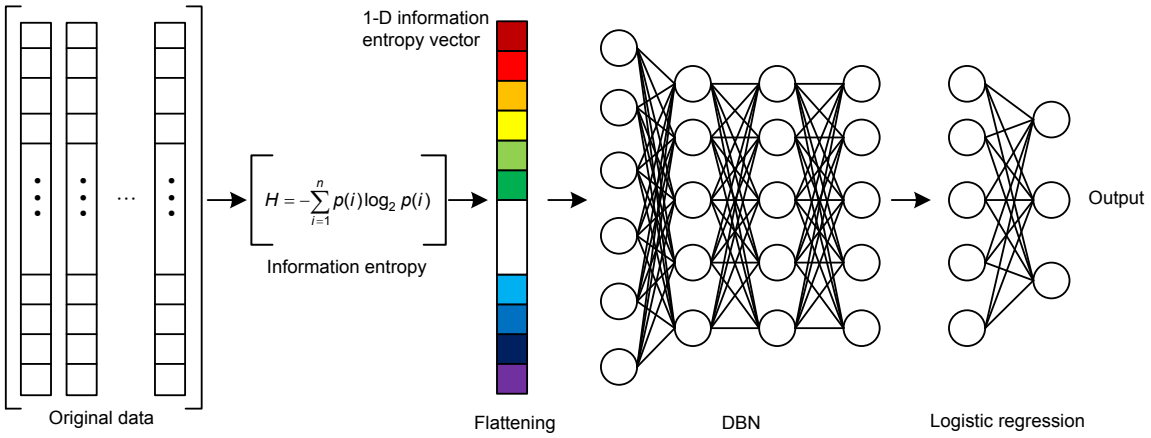

Fig. 6 Gas turbine engine fault diagnosis using deep belief networks (DBNs) based on information entropy and a logistic regression method

of DBN. The outputs of the logistic regression layer have three neurons. The logistic regression layer is used to fine-tune the pre-trained deep belief network and combine the output features of the network in engine fault classification.

The first step in the IE-DBNs method is to calculate the information entropy from the original data. The outputs from sensors and test equipment are the original data, such as temperature, speed, and vibration.

Then, the calculated information entropies will be the input vectors $\boldsymbol{v}$ of DBNs, and the features of the input data are trained by DBNs. The information entropy vector could reflect the energy distribution of the original data and the nonlinear feature of the system. At the same time, the information entropy, as the pre-training in DBNs, will reduce the dimension of the input into DBNs and improve the accuracy of the output of DBNs.

Furthermore, the training process for DBNs is layer by layer. For example, we train the first layer and second layer first, and then train the second layer and third layer.

At last, we can obtain the results of fault diagnosis through the logistic regression layer processing of fault labels and learned features. The joint complexity information entropy and DBN are skilled in processing nonlinear signals and reconstructing the complex systems. Thus, the running time and diagnostic accuracy of IE-DBNs will be better than those of traditional supervised methods, which will be verified in the next section.

4 Simulations

4.1 Simulation description

The proposed multi-sensor fault diagnosis for a gas turbine engine using the IE-DBNs method is demonstrated with the data generated from the gas turbine engine thermo-dynamical simulation. By thermo-dynamical gas turbine engine model simulation, we can simulate a real aircraft turbofan engine. The engine model considers the volume dynamics, damage factors, and adds in an unbalanced mass flow rate. In the simulation there are 14 different inputs, and the outputs contain data for 21 different sensor measurements. The inputs enable the user to change the amplitude of variation which is used to simulate the fault process containing the fan, low-pressure compressor (LPC), and high-pressure compressor (HPC). The dataset from the output of the gas turbine engine thermo-dynamical simulated sensors is composed of engine operating data from a normal state to a failure state. Each engine is of the same type, but the initial wear, product deviation, and degradation rate may be different. Each engine works normally at the beginning, and some design fault degradations start to appear at some point, and then the engine stops working when it breaks down. In this simulation, we chose 250 engine samples as the operating dataset to demonstrate the engine fault diagnosis based on the IE-DBNs method. The performances of fault diagnosis based on the IE-DBNs method for engine operating datasets are presented in the following.

4.2 Fault mode definition and information entropy (data preprocessing)

The engine fault-mode definition is detailed in Table 1. Four fault modes were used in this simulation, each dataset containing one fault mode. Furthermore, in the gas turbine engine thermo-dynamical simulation model, the four fault modes are not sudden failures; instead, they are all degradation failures.

Table 1 Engine fault mode definition

No.	Fault mode	Fault code
1	Fan failure	001
2	Low pressure compressor failure	010
3	High pressure compressor failure	100
4	Fan and high pressure compressor failure	110

Fig. 7 shows the six steps for sensory data preprocessing. When dealing with an actual problem, we cannot install too many sensors in the engine machine. Thus, we chose four sensors among the 21 sensors, which tallies with the actual situation. These four sensors were used to monitor the temperature of the LPC outlet, the temperature of the HPC outlet, the physical fan speed, and the physical core speed, respectively. That is to say, we used four sensory datasets to diagnose the engine fault mode. Fig. 8 displays the total temperature at the LPC outlet from engine 1 to engine 5, and Fig. 9 indicates the HPC outlet temperature.

The physical core speed from the engine simulation result is shown in Fig. 10. We cannot see the

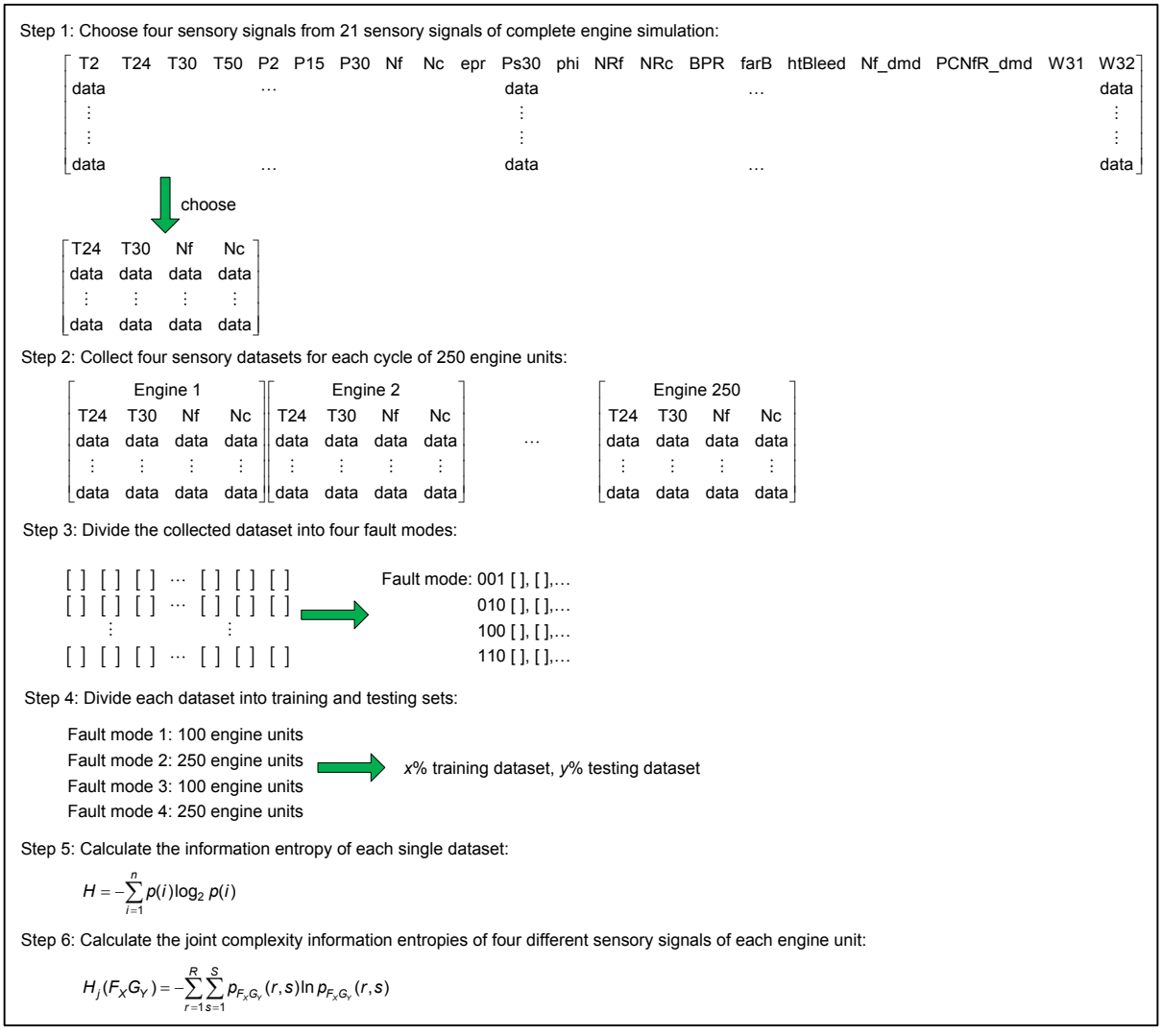

Fig. 7 Procedure for preprocessing the engine simulation data

time when the degradation begins and the fault does not appear clearly from the raw data. However, from a careful consideration of Figs. 6, 9, and 10, readers can see that each sample can be regarded as increasing all the time cycles, and when the physical core speed increases, the temperature of HPC and LPC can increase at the same time.

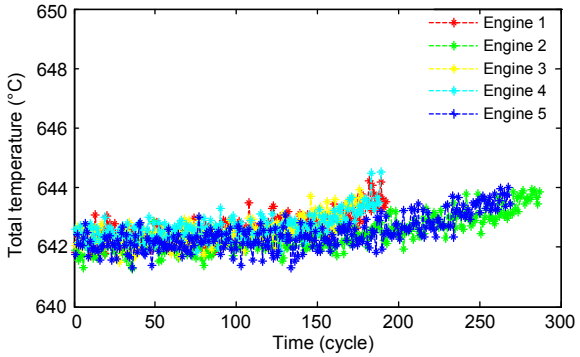

Fig. 8 The simulation results from a gas turbine engine for the total temperature at the low-pressure compressor

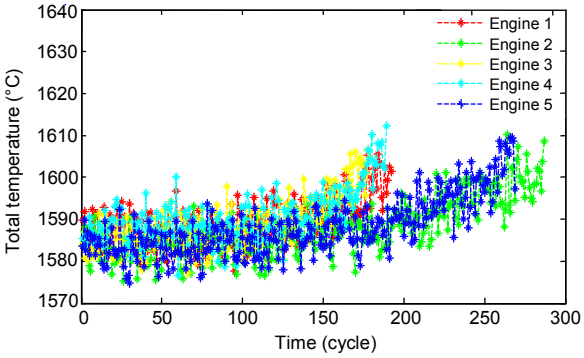

Fig. 9 The simulation results from a gas turbine engine for the total temperature at the high-pressure compressor

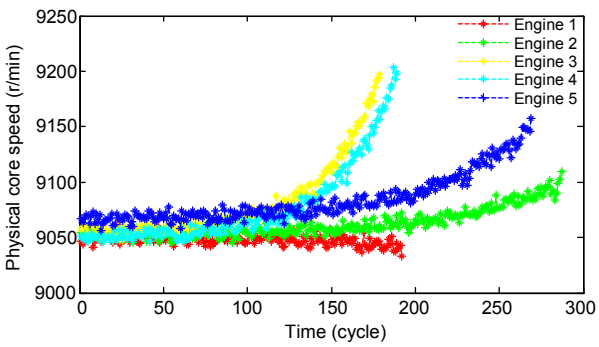

Fig. 10 The simulation results from a gas turbine engine for the physical core speed

The output of the simulated model for each engine contains operating data for one fault mode, and the data series ends as the engine lifecycle ends. All the selected datasets were divided into training datasets and testing datasets (Table 2). Then, the training datasets will be trained separately by the IE-DBNs model.

Table 2 Three different experiments

Experiment	Percentage of data	
	Training	Testing
E1	40%	60%
E2	50%	50%
E3	60%	40%

Tables 3–7 show the information entropy features of the training datasets with the different fault modes. These tables are joint complexity information entropy based on singular spectrum, power spectrum, wavelet energy spectrum, multi-resolution spectrum, and multi-resolution singular spectrum, respectively. Each table lists four samples, which represent the different fault modes. In Table 3, $H_{s(T24)}$, $H_{s(T30)}$, $H_{s(Nf)}$, and $H_{s(Nc)}$ are the singular spectrum entropy of four sensory signals, respectively; $H_{js(T24\text{-}T30)}$, $H_{js(T24\text{-}Nf)}$, $H_{js(T24\text{-}Nc)}$, $H_{js(T30\text{-}Nf)}$, $H_{js(T30\text{-}Nc)}$, and $H_{js(Nf\text{-}Nc)}$ are the joint complexity information entropies based on the singular spectrum between the four sensory signals. Additionally, the data from Tables 4–7 follows the same format as Table 3. $\boldsymbol{P}=(p_1, p_2, p_3)$ is the state space of the sample.

4.3 Fault diagnosis based on IE-DBNs

$\boldsymbol{H}=[H_{s(T24)}, H_{s(T30)}, H_{s(Nf)}, H_{s(Nc)}, H_{js(T24\text{-}T30)}, H_{js(T24\text{-}Nf)}, H_{js(T24\text{-}Nc)}, H_{js(T30\text{-}Nf)}, H_{js(T30\text{-}Nc)}, H_{js(Nf\text{-}Nc)}]$ is considered the input for the DBNs. $\boldsymbol{P}=[p_1, p_2, p_3]$ is the output of the DBNs. We used 100 entropy samples, similar to Table 3, for training and learning in the DBNs through the proposed approach described in Section 3.3. The trained IE-DBNs has one input layer (which includes 10 neurons), one output layer (which includes three neurons), and four hidden layers (where each layer has 50 neurons). Then, we can obtain the optimum structure and weight for the DBNs. We also used state-of-the-art machine learning methods to diagnose the engine fault, for comparison with the IE-DBNs method (Figs. 11 and 12). The machine learning methods include DBN, binary

Table 3 Joint complexity information entropy based on a singular spectrum

Sample	$H_{s(T24)}$	$H_{s(T30)}$	$H_{s(Nf)}$	$H_{s(Nc)}$	$H_{js(T24\text{-}T30)}$	$H_{js(T24\text{-}Nf)}$	$H_{js(T24\text{-}Nc)}$	$H_{js(T30\text{-}Nf)}$	$H_{js(T30\text{-}Nc)}$	$H_{js(Nf\text{-}Nc)}$	P
1	38.532	37.436	38.921	36.854	60.354	58.652	40.568	59.457	61.436	62.662	001
2	35.896	37.889	37.854	37.568	59.635	57.698	45.364	56.125	59.634	58.621	010
3	36.457	35.478	36.889	37.486	55.561	60.248	48.328	60.654	58.754	59.324	100
4	30.784	31.325	32.554	31.869	53.254	55.325	40.215	53.654	49.194	51.984	110

Table 4 Joint complexity information entropy based on a power spectrum

Sample	$H_{p(T24)}$	$H_{p(T30)}$	$H_{p(Nf)}$	$H_{p(Nc)}$	$H_{jp(T24\text{-}T30)}$	$H_{jp(T24\text{-}Nf)}$	$H_{jp(T24\text{-}Nc)}$	$H_{jp(T30\text{-}Nf)}$	$H_{jp(T30\text{-}Nc)}$	$H_{jp(Nf\text{-}Nc)}$	P
1	29.657	26.355	24.658	27.824	51.328	48.623	41.365	47.354	49.365	48.326	001
2	26.869	27.634	26.634	23.365	50.148	49.398	40.487	48.361	49.447	50.871	010
3	27.634	23.653	24.318	28.614	45.614	46.874	39.654	54.369	51.485	55.985	100
4	20.634	19.588	18.346	19.676	46.365	43.489	38.364	40.315	41.354	42.314	110

Table 5 Joint complexity information entropy based on a wavelet energy spectrum

Sample	$H_{w(T24)}$	$H_{w(T30)}$	$H_{w(Nf)}$	$H_{w(Nc)}$	$H_{jw(T24\text{-}T30)}$	$H_{jw(T24\text{-}Nf)}$	$H_{jw(T24\text{-}Nc)}$	$H_{jw(T30\text{-}Nf)}$	$H_{jw(T30\text{-}Nc)}$	$H_{jw(Nf\text{-}Nc)}$	P
1	21.489	20.364	19.354	18.346	36.354	39.154	32.345	40.125	41.398	40.694	001
2	20.364	21.365	18.487	17.469	35.647	38.376	33.654	35.315	39.634	37.861	010
3	20.746	21.914	17.634	16.984	37.614	36.354	31.417	39.364	42.315	40.694	100
4	15.348	14.698	13.654	14.623	28.647	29.315	25.614	27.314	30.654	29.637	110

Table 6 Joint complexity information entropy based on a multi-resolution spectrum

Sample	$H_{r(T24)}$	$H_{r(T30)}$	$H_{r(Nf)}$	$H_{r(Nc)}$	$H_{jr(T24\text{-}T30)}$	$H_{jr(T24\text{-}Nf)}$	$H_{jr(T24\text{-}Nc)}$	$H_{jr(T30\text{-}Nf)}$	$H_{jr(T30\text{-}Nc)}$	$H_{jr(Nf\text{-}Nc)}$	P
1	58.523	55.954	59.364	57.886	84.361	88.648	80.945	89.374	87.614	88.598	001
2	56.658	54.659	57.154	58.945	89.314	90.564	82.751	88.945	88.197	86.647	010
3	54.634	57.648	58.973	56.841	87.648	86.454	81.452	89.784	90.841	87.548	100
4	50.485	51.861	48.945	49.713	80.751	79.648	75.186	81.647	82.191	81.649	110

Table 7 Joint complexity information entropy based on a multi-resolution singular spectrum

Sample	$H_{rs(T24)}$	$H_{rs(T30)}$	$H_{rs(Nf)}$	$H_{rs(Nc)}$	$H_{jrs(T24\text{-}T30)}$	$H_{jrs(T24\text{-}Nf)}$	$H_{jrs(T24\text{-}Nc)}$	$H_{jrs(T30\text{-}Nf)}$	$H_{jrs(T30\text{-}Nc)}$	$H_{jrs(Nf\text{-}Nc)}$	P
1	64.842	68.774	66.155	69.633	95.126	97.487	90.753	96.951	97.741	98.852	001
2	62.963	66.135	65.351	68.576	94.378	98.667	91.512	97.314	98.548	96.549	010
3	63.794	67.156	64.787	66.359	96.518	97.488	93.367	95.699	96.487	94.124	100
4	58.314	59.641	60.459	56.259	86.147	87.315	90.546	88.149	89.649	90.181	110

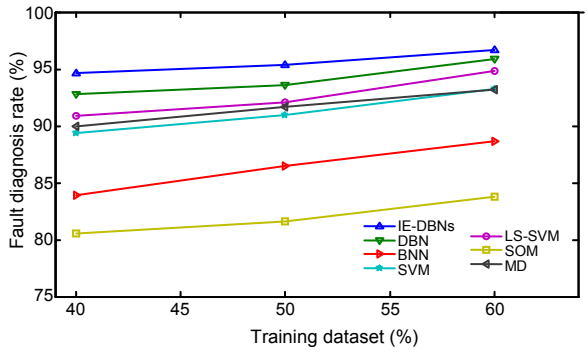

Fig. 11 Fault diagnosis results from the training

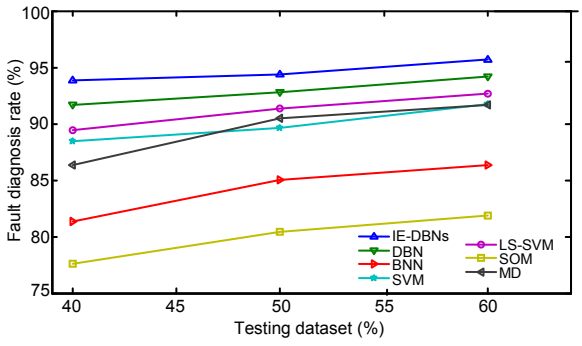

Fig. 12 Fault diagnosis results from the testing

neural networks (BNN), SVM, least squares support vector machine (LS-SVM), self-organizing map (SOM), and morphology discrete wavelet transform (MD).

The structure of each fault diagnosis model used for comparison is as follows: The structure of the BNN has three processing layers, input, hidden, and output, with 7, 5, and 3 neurons in each layer, respectively. Tanh was used as the transfer function. A Gaussian kernel function was used to train the one-against-all SVM fault diagnosis model. The training architecture of SOM has 10×10 neurons. The training and testing datasets were used in all of the methods above, and the testing datasets were used to verify the fault accuracy of these methods.

Three different experiments were conducted to evaluate the performance of the IE-DBNs model. Table 2 shows the results. E1 has 40% of training datasets and 60% of testing datasets; the percentages of training datasets and testing datasets are equal in E2; and E3 is the inverse of E1. The results are listed in Figs. 11–14.

4.4 Results and discussion

The results are listed in Tables 8 and 9. In Table 8, $H_{s(T24)}$, $H_{s(T30)}$, $H_{s(Nf)}$, and $H_{s(Nc)}$ are the singular

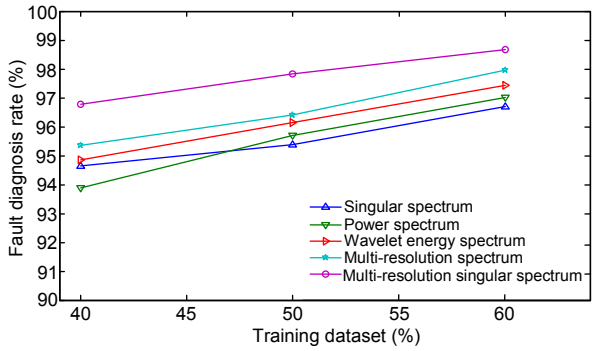

Fig. 13 Fault diagnosis results using the IE-DBNs method based on the five different information entropies from the training

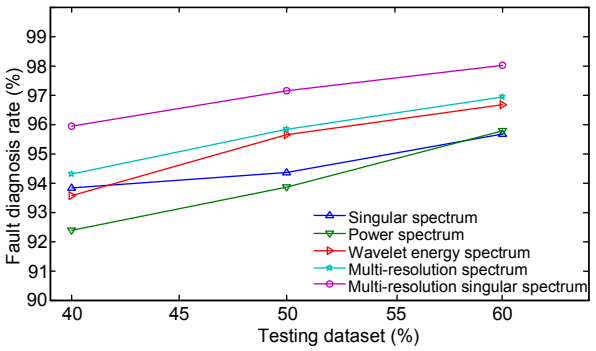

Fig. 14 Fault diagnosis results using the IE-DBNs method based on the five different information entropies from the testing

Table 8 Fault diagnosis results using IE-DBNs based on the five different information entropies

Information entropy	Fault diagnosis rate (%)					
	Training			Testing		
	E1	E2	E3	E1	E2	E3
H_s	95.37	94.65	96.71	94.35	93.84	95.68
H_p	95.69	93.88	97.01	93.85	92.39	95.78
H_w	96.15	94.86	97.43	95.64	93.57	96.67
H_r	96.41	95.35	97.96	95.84	94.31	96.94
H_{rs}	97.82	96.77	98.68	97.14	95.93	98.01

Table 9 Fault diagnosis results obtained using the proposed IE-DBNs method and some existing methods

Method	Fault diagnosis rate (%)					
	Training			Testing		
	E1	E2	E3	E1	E2	E3
IE-DBNs	**95.37**	**94.65**	**96.71**	**94.35**	**93.84**	**95.68**
DBN	93.56	92.78	95.89	92.81	91.69	94.16
BNN	86.47	83.89	88.68	85.02	81.34	86.37
SVM	90.96	89.36	93.25	89,64	88.48	91.74
LS-SVM	92.10	90.88	94.86	91.33	89.45	92.65
SOM	81.58	80.54	83.78	80.45	77.63	81.87
MD	91.69	89.96	93.22	90.47	86.34	91.68

spectrum entropies of the four sensory signals, respectively; $H_{js(T24\text{-}T30)}$, $H_{js(T24\text{-}Nf)}$, $H_{js(T24\text{-}Nc)}$, $H_{js(T30\text{-}Nf)}$, $H_{js(T30\text{-}Nc)}$, and $H_{js(Nf\text{-}Nc)}$ are the joint complexity information entropies based on the singular spectrum between the four sensory signals.

The diagnosis results for E1, E2, and E3 in Figs. 11 and 12 show that the different numbers of training samples will produce different levels of accuracy. We can clearly see that the fault diagnosis rates of all the algorithms are decreasing in the following order: E3, E1, E2. A simple explanation is that the more training samples there are, the higher the precision will be. Compared with the six existing machine learning approaches for diagnosis, the correct diagnosis rate of the proposed IE-DBNs method is higher than those of the other methods. For example, the diagnosis rates for 50% of training datasets are 95.37%, 93.56%, 86.47%, 90.96%, 92.10%, 81.58%, and 91.69% for the IE-DBNs, DBN, BNN, SVM, LS-SVM, SOM, and MD methods, respectively (Fig. 11). The main reason for this is that DBNs has a very strong ability in learning and describing signal complexity, which has nonlinear relationships between the input sensory signals and the different fault states. Information entropy is also good at reflecting the nonlinear relationships between signals and it also contains the system energy feature. Additionally, DBN has easy encoding for richer and higher-order network structures within the deep learning process for both supervised and unsupervised training. It is also obvious that the fault diagnosis rate of the SOM method is lower than those of the other methods, because of its low efficiency in learning nonlinear, complex signals. Thus, the IE-DBNs method demonstrates excellent performance in dealing with complex and nonlinear problems.

Figs. 13 and 14 show the diagnosis results for the five different information entropies from IE-DBNs. For example, Fig. 13 shows that the diagnosis rates for 50% of training datasets are 95.37%, 95.69%, 96.15%, 96.41%, and 97.82% for singular spectrum, power spectrum, wavelet energy spectrum, multi-resolution spectrum, and multi-resolution singular spectrum, respectively. Clearly, the correct diagnosis rate for multi-resolution singular spectrum entropy with input from IE-DBNs is the highest among the five entropies. Because multi-resolution singular spectrum entropy has better nonlinear feature extraction and noise suppression capabilities than the other four entropies, H_{rs} will give DBNs a feature with a clear structure that is easy to learn. As a result, the proposed IE-DBNs method has an outstanding performance in diagnosing engine faults, and this is especially true for information entropy derived from multi-resolution singular spectrum entropy. The limitation of the proposed methodology is the requirement of a large dataset to calculate the information entropy. If we had not tested sufficient data, we could not have used this method.

5 Conclusions

We have proposed the IE-DBNs method and provided some contributions to the domain of engine fault diagnosis. Information entropy is used in fault diagnosis and constitutes the input for deep belief networks. Joint complexity information entropy is proposed as a way to process multi-sensor signals and describe a nonlinear feature between two signals. We can obtain the most suitable features for signals through joint complexity. Furthermore, deep belief networks have a strong ability to learn complex systems. Thus, IE-DBNs is a promising method to diagnose gas turbine engine faults that occur at high temperature, high pressure, high rotating speeds, and in harsh environments. From the results, the accuracy of IE-DBNs is higher than that of existing methods, such as SVM, BNN, and DBN. This is especially true for the multi-resolution singular spectrum approach, because in this case IE-DBNs can clearly extract features and easily learn the structure of the training data, thus having the best diagnostic performance compared to the other entropies. As a result, the proposed IE-DBNs algorithm can provide precise yet robust results for gas turbine engines where there are complex system fault diagnosis applications, and in particular for systems that have highly nonlinear relationships between the inputs and the fault diagnosis state outputs. In further study we may consider unlabeled data learning, unknown engine faults, and the application of the method to real engine machines.

References

Aguiar, V., Guedes, I., 2015. Shannon entropy, Fisher information and uncertainty relations for log-periodic oscillators. *Phys. A*, **423**:72-79. http://dx.doi.org/10.1016/j.physa.2014.12.031

Bengio, Y., 2009. Learning Deep Architectures for AI. Available from http://www.iro.umontreal.ca/~bengioy/papers/ftml.pdf

Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. *LNCS*, **7700**:437-478. http://dx.doi.org/10.1007/978-3-642-35289-8_26

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: a review and new perspectives. *IEEE Trans. Patt. Anal. Mach. Intell.*, **35**(8):1798-1828. http://dx.doi.org/10.1109/TPAMI.2013.50

Bottou, L., 2012. Stochastic gradient descent tricks. *LNCS*, **7700**:421-436. http://dx.doi.org/10.1007/978-3-642-35289-8_25

Chen, Y.S., Zhao, X., Jia, X.P., 2015. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, **8**(6):2381-2392. http://dx.doi.org/10.1109/JSTARS.2015.2388577

Cui, H.X., Zhang, L.B., Kang, R.Y., *et al*., 2009. Research on fault diagnosis for reciprocating compressor valve using information entropy and SVM method. *J. Loss Prevent. Process Ind.*, **22**(6):864-867. http://dx.doi.org/10.1016/j.jlp.2009.08.012

Dai, J.H., Tian, H.W., 2013. Entropy measures and granularity measures for set-valued information systems. *Inform. Sci.*, **240**:72-82. http://dx.doi.org/10.1016/j.ins.2013.03.045

Ferrer, A., 2007. Multivariate statistical process control based on principal component analysis (MSPC-PCA): some reflections and a case study in an autobody assembly process. *Qual. Eng.*, **19**(4):311-325. http://dx.doi.org/10.1080/08982110701621304

Geng, J.B., Huang, S.H., Jin, J.S., *et al*., 2006. A method of rotating machinery fault diagnosis based on the close degree of information entropy. *Int. J. Plant Eng. Manag.*, **11**(3):137-144. http://dx.doi.org/10.13434/j.ckni.1007-4546.2006.03.002

Hinton, G.E., 2010. A Practical Guide to Training Restricted Boltzmann Machines. Available from https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf

Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neur. Comput.*, **18**(7):1527-1554. http://dx.doi.org/10.1162/neco.2006.18.7.1527

Hinton, G.E., Deng, L., Yu, D., *et al.*, 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.*, **29**(6):82-97. http://dx.doi.org/10.1109/MSP.2012.2205597

Jin, C.X., Li, F.C., Li, Y., 2014. A generalized fuzzy ID3 algorithm using generalized information entropy. *Knowl.-Based Syst.*, **64**:13-21. http://dx.doi.org/10.1016/j.knosys.2014.03.014

Koverda, V.P., Skokov, V.N., 2012. Maximum entropy in a nonlinear system with a 1/*f* power spectrum. *Phys. A*, **391**(1-2):21-28. http://dx.doi.org/10.1016/j.physa.2011.07.015

Larochelle, H., Bengio, Y., Louradour, J., *et al*., 2009. Exploring strategies for training deep neural networks. *J. Mach. Learn. Res.*, **10**(10):1-40.

Li, F.C., Zhang, Z., Jin, C.X., 2016. Feature selection with partition differentiation entropy for large-scale data sets. *Inform. Sci.*, **329**:690-700. http://dx.doi.org/10.1016/j.ins.2015.10.002

Li, J., 2015. Recognition of the optical image based on the wavelet space feature spectrum entropy. *Optik-Int. J. Light Electron Opt.*, **126**(23):3931-3935. http://dx.doi.org/10.1016/j.ijleo.2015.07.166

Liu, Z.G., Hu, Q.L., Cui, Y., *et al*., 2014. A new detection approach of transient disturbances combining wavelet packet and Tsallis entropy. *Neurocomputing*, **142**:393-407. http://dx.doi.org/10.1016/j.neucom.2014.04.020

Martens, J., Sutskever, I., 2012. Training deep and recurrent networks with Hessian-free optimization. *LCNS*, **7700**:479-535. http://dx.doi.org/10.1007/978-3-642-35289-8_27

Memisevic, R., Hinton, G.E., 2010. Learning to represent spatial transformations with factored higher-order Boltzmann machine. *Neur. Comput.*, **22**(6):1473-1492. http://dx.doi.org/10.1162/neco.2010.01-09-953

Mohamed, A.R., Dahl, G.E., Hinton, G.E., 2012. Acoustic modeling using deep belief networks. *IEEE Trans. Audio Speech Lang. Process.*, **20**(1):14-22. http://dx.doi.org/10.1109/TASL.2011.2109382

Nichols, J.M., Seaver, M., Trickey, S.T., 2006. A method for detecting damage-induced nonlinearities in structures using information theory. *J. Sound Vibr.*, **297**(1-2):1-16. http://dx.doi.org/10.1016/j.jsv.2006.01.025

Niu, J., Bu, X.Z., Li, Z., *et al*., 2014. An improved bilinear deep belief network algorithm for image classification. 10th Int. Conf. on Computational Intelligence and Security, p.189-192. http://dx.doi.org/10.1109/CIS.2014.38

Nourani, V., Alami, M.T., Vousoughi, F.D., 2015. Wavelet-entropy data pre-processing approach for ANN-based groundwater level modeling. *J. Hydrol.*, **524**:255-269. http://dx.doi.org/10.1016/j.jhydrol.2015.02.048

Ong, B.T., Sugiura, K., Zettsu, K., 2014. Dynamic pre-training of deep recurrent neural networks for predicting environmental monitoring data. IEEE Int. Conf. on Big Data, p.760-765. http://dx.doi.org/10.1109/BigData.2014.7004302

Pan, Y.B., Yang, B.L., Zhou, X.W., 2015. Feedstock molecular reconstruction for secondary reactions of fluid catalytic cracking gasoline by maximum information entropy method. *Chem. Eng. J.*, **281**:945-952. http://dx.doi.org/10.1016/j.cej.2015.07.037

Rastegin, A.E., 2015. On generalized entropies and information-theoretic Bell inequalities under decoherence. *Ann. Phys.*, **355**:241-257. http://dx.doi.org/10.1016/j.aop.2015.02.015

Rodríguez, P.H., Alonso, J.B., Ferrer, M.A., *et al*., 2013. Application of the Teager-Kaiser energy operator in bearing

fault diagnosis. *ISA Trans.*, **52**(2):278-284. http://dx.doi.org/10.1016/j.isatra.2012.12.006

Saimurugan, M., Ramachandran, K.I., Sugumaran, V., *et al*., 2011. Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine. *Expert Syst. Appl.*, **38**(4):3819-3826. http://dx.doi.org/10.1016/j.eswa.2010.09.042

Sainath, T.N., Kingsbury, B., Soltau, H., *et al*., 2013. Optimization techniques to improve training speed of deep neural networks for large speech tasks. *IEEE Trans. Audio Speech Lang. Process.*, **21**(11):2267-2276. http://dx.doi.org/10.1109/TASL.2013.2284378

Sainath, T.N., Kingsbury, B., Saon, G., *et al*., 2015. Deep convolutional neural networks for large-scale speech tasks. *Neur. Networks*, **64**:39-48. http://dx.doi.org/10.1016/j.neunet.2014.08.005

Sekerka, R.F., 2015. Entropy and information theory. *In*: Thermal Physics: Thermodynamics and Statistical Mechanics for Scientists and Engineers. Elsevier, p.247-256. http://dx.doi.org/10.1016/B978-0-12-803304-3.00015-6

Sermanet, P., Chintala, S., LeCun, Y., 2012. Convolutional neural networks applied to house numbers digit classification. 21st Int. Conf. on Pattern Recognition, p.3288-3291.

Song, X.D., Sun, G.H., Dong, S.H., 2015. Shannon information entropy for an infinite circular well. *Phys. Lett. A*, **379**(22-23):1402-1408. http://dx.doi.org/10.1016/j.physleta.2015.03.020

Su, H.T., You, G.J.Y., 2014. Developing an entropy-based model of spatial information estimation and its application in the design of precipitation gauge networks. *J. Hydrol.*, **519**(D):3316-3327. http://dx.doi.org/10.1016/j.jhydrol.2014.10.022

Susan, S., Hanmandlu, M., 2013. A non-extensive entropy feature and its application to texture classification. *Neurocomputing*, **120**:214-225. http://dx.doi.org/10.1016/j.neucom.2012.08.059

Sutskever, I., Hinton, G.E., Taylor, G.W., 2008. The recurrent temporal restricted Boltzmann machine. Proc. 22nd Annual Conf. on Neural Information Processing Systems, p.1601-1608.

Tamilselvan, P., Wang, P.F., 2013. Failure diagnosis using deep belief learning based health state classification. *Reliab. Eng. Syst. Safety*, **115**:124-135. http://dx.doi.org/10.1016/j.ress.2013.02.022

Tamilselvan, P., Wang, P.F., Youn, B.D., 2011. Multi-sensor health diagnosis using deep belief network based state classification. ASME Int. Design Engineering Technical Conf. & Computers and Information in Engineering Conf., p.749-758. http://dx.doi.org/10.1115/DETC2011-48352

Tran, V.T., AlThobiani, F., Ball, A., 2014. An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks. *Expert Syst. Appl.*, **41**(9):4113-4122. http://dx.doi.org/10.1016/j.eswa.2013.12.026

Xie, Y., Zhang, T., 2005. A fault diagnosis approach using SVM with data dimension reduction by PCA and LDA method. Chinese Automation Congress, p.869-874. http://dx.doi.org/10.1109/CAC.2015.7382620

Zhang, W.L., Li, R.J., Deng, H.T., *et al*., 2015. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage*, **108**:214-224. http://dx.doi.org/10.1016/j.neuroimage.2014.12.061

Zhao, X.Z., Ye, B.Y., 2016. Singular value decomposition packet and its application to extraction of weak fault feature. *Mech. Syst. Signal Process.*, **70-71**:73-86. http://dx.doi.org/10.1016/j.ymssp.2015.08.033

Zhou, S.S., Chen, Q.C., Wang, X.L., 2014. Deep adaptive networks for visual data classification. *J. Multim.*, **9**(10): 1142-1151.