Applied Intelligence
https://doi.org/10.1007/s10489-022-04342-1

An effective zero-shot learning approach for intelligent fault detection using 1D CNN

Siyu Zhang[1,2] · Hua-Liang Wei[1] · Jinliang Ding[2]

Accepted: 11 November 2022

Abstract
Data-driven fault detection techniques have attracted extensive attention in engineering, industry and many other areas in recent years. In many real applications, the following situation often occurs: data for certain types of faults (unseen faults) are not available to train models that are used for fault detection. Such a scenario can occur when data collection becomes highly time-consuming or destructive. To address this challenging problem, a novel fault detection method using zero-shot learning (ZSL) is proposed in this paper, which contains three phases: feature extraction, label embedding, and feature embedding. The method first extracts features from raw signals by applying a one-dimensional convolutional neural network (1D CNN), then builds semantic descriptions (human-defined) as fault attributes shared between seen faults and unseen faults, and finally uses a bi-linear compatibility function to find the highest-ranking fault type. The proposed semantic space based zero-shot learning with 1D CNN is called SSB-ZSL-1DCNN. The cosine distance is used to measure the similarity between feature embeddings and fault attributes. An important characteristic of SSB-ZSL-1DCNN is that the model, trained using only samples of seen faults, can be used to detect unseen defects. To evaluate the proposed method, two case studies are designed based on two well-known benchmarks (the Tennessee-Eastman chemical control process and the rolling bearing experiments at the Case Western Reserve University, respectively). The results demonstrate that the proposed method shows remarkable performance in detecting unseen faults.

Keywords Fault detection · Zero-shot learning · Deep learning · Semantic description · Label embedding · Convolutional neural network

1 Introduction

Fault detection and classification play an essential role in process control, monitoring, health management and maintenance because it builds a bridge between system monitoring data and its health status [1]. In industry, many complex process systems or plants comprise a large number of components; many key components (e.g. rolling bearings commonly used in machines) are expensive, vulnerable to damage, and prone to fault. Therefore, carrying out condition monitoring and fault detection of complex processes and machines are paramount for either safety or economic purposes.

Traditional fault detection and diagnosis methods can be roughly categorized into two groups: model-based and data-driven methods. The former relies on explicit mathematical models of the plant, while the latter uses historical data of the plant to determine its health status [2]. Traditional methods work under the following assumption: the distribution of the training data is similar to that of the test data (e.g., the fault samples), implying that the training data should contain a good number of fault samples so that the models used for fault detection are well trained. However, such an assumption may be violated in many real applications due to the following reasons. Firstly, it usually assumes that data are well collected and sufficiently represent both health and all potential faults status of the system (machine, plant, equipment, etc.) of interest. In practice, however, it could be such a case where no or few samples of target faults are available during system operation [3]. Secondly, few plants or systems would be allowed to operate to an occurrence of a major fault or a number of minor faults; therefore, normally it is very difficult (if not impossible) to collect a sufficiently large amount of samples to well

✉ Hua-Liang Wei
w.hualiang@sheffield.ac.uk

[1] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK

[2] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China

Published online: 01 December 2022

train a fault detection model, because faults are destructive and can result in enormous losses [4]. Thirdly, systems (e.g. machines) typically gradually decline from health to failure, implying that obtaining adequate fault samples for data-driven approaches is time-consuming and costly.

A way to effectively solve fault detection tasks with small and imbalanced data is to develop intelligent fault detection (IFD) algorithms using, e.g., data augmentation-based strategy and transfer learning [1]. Such an algorithm first augments the data-by-data generation or data resampling, then uses the augmented data to extract features with machine learning models, e.g., neural networks, together with a feature adaptation process (where necessary), and finally builds a suitable fault classifier to identify the types of faults. However, in practice there exists such a case where a certain type of fault may not be observed or recorded due to some reasons.

In certain extreme circumstances, signals for specific fault types or working conditions are unobtainable, implying that diagnosis models cannot be well trained due to the lack of training samples for unseen fault classes. Moreover, in data-driven fault detection, identifying the unseen fault classes is a challenging task for traditional IFD methods [1]. Still, in practice there is a high need to tackle a realistic and highly challenging scenario as follows: samples of one or more certain fault types are not available at all or just a very limited small number of samples are available. In recent years, a learning method, called zero-shot learning (ZSL), has been widely used in image classification due to its power to recognize new objects (not seen in the model training stage) based on information inferred from seen classes [5]. ZSL provides a powerful tool for solving the unseen fault detection problem concerned in this work, and a reasonable solution is to combine fault detection with zero-shot learning to classify the certain (unseen) types of faults without using samples of these types of faults. Inspired by the idea of zero-shot learning [6], this paper proposes a new zero-shot fault detection method based on a semantic space embedded model for industrial systems or devices. The implementation of the proposed method is as follows. Step 1: To build convolutional neural networks for feature extraction from raw data; Step 2: To define and specify a semantic based space for faults by creating a shared attribute form (matrix) for each type of fault; Step 3: To adopt and define a bilinear compatibility function to learn the relationship between the extracted features and fault attributes, based on which the highest-ranking unseen fault class is determined.

The performance of the proposed method is tested and assessed on real datasets, collected from a bearing and a complex chemical system, respectively. Two case studies are presented accordingly. The first case is about the bearing, widely used as a piece of crucial rotary equipment in many applications. We propose a method to detect 'large-diameter faults' by training classifiers only using samples of 'small-diameter faults'. The second case emphasizes an entire and comprehensive industrial process control system, which aims to detect different unseen types of faults (i.e., not used in the training stage).

The main contributions of this paper are summarized as follows:

1) We design a new zero-shot learning scheme for unseen fault detection without using samples of the unseen faults in the training stage. Specifically, for the bearing case, the proposed approach can detect and identify large-diameter in bearing, by training a model using data containing small-diameter fault samples but without including samples of large-diameter faults.
2) To improve the adaptation of zero-shot learning to 1D time-series (most industrial fault data are in such a format), we comprehensively analyze various feature extraction methods, including traditional methods and deep learning based methods.
3) The proposed semantic space, which treats the fault attributes as the side information, builds a bridge from seen faults to unseen faults for zero-shot fault detection.

The remainder of the paper is organized as follows. Section 2 provides a relevant literature review on traditional, intelligent, and zero-shot fault detection. Section 3 presents the details of our proposed method. Section 4 conducts two case studies of fault detection to demonstrate and verify the effectiveness of the proposed method. Case 1 is concerned with the detection of the unseen faults of larger size diameters occurring in the Case Western Reserve University (CWRU) experimental dataset. Case 2 conducts experiments on the Tennessee-Eastman process (TEP) dataset, aiming at detecting unseen types of faults. Finally, Section 5 concludes the main work.

2 Relevant literate review

2.1 Traditional fault detection and intelligent fault detection

The procedure of traditional classification-based, data-driven fault detection contains three main steps: data acquisition, feature extraction, and fault detection and classification [7]. In practice, data are collected via different means including the use of numerous sensors. Feature extraction is usually implemented through linear or nonlinear transformation and data decomposition. Commonly used linear methods includes principal component analysis (PCA) [8] and independent component analysis (ICA) [9]. Nonlinear data processing approaches, such as kernel based methods, are usually more powerful for characterizing nonlinear relationships, for example, it has been proved that the kernel principal analysis (KPCA) works better than its linear counterpart PCA for many applications [10]. It is usually important and useful to reduce the dimensionality of features or variables in the training space for several reasons, e.g., to make the classification tasks easier to implement, improve classification accuracy, or make the models and results easier to explain. PCA performs poorly when extracting features from a set of signals that are nonlinearly associated to or dependent on each other. ICA can work better for extracting non-Gaussian features from multivariate signals [11].

Recently, 1D convolutional neural network (CNN) was introduced to extract optimal damage-sensitive features automatically for vibration-based fault detection. For example, in [12], raw signals were transformed into two-dimensional grayscale images based on wavelet transform and deep CNN to extract robust features. In [13], a novel fault detection and classification method was proposed by using the DWT and CWT filter banks.

After feature extraction, the resulting features are fed into a fault classification model to determine the system's health status. Many machine learning models have been developed for fault detection and classification (see e.g., [14]). In [15], a bearing fault diagnosis method was proposed based on deep CNN and random forest (RF) ensemble learning.

As mentioned in Section 1, traditional fault detection methods may not work well with small or imbalance data, therefore intelligent fault detection methods are needed to guarantee fault detection performance. A way to build intelligent fault detectors is to use deep learning. Deep transfer learning (DTL) methods have been introduced to the field of fault detection to overcome the difficulty in data collection (e.g., samples of certain faults are not enough or not available). DTL methods treat the insufficient samples as a cross-domain learning task and aims to find a solution by performing domain adaptation [16] for handling different distributions between source domain data and target domain data. Zhang et al. [17] investigated an end-to-end method based on a deep convolutional neural network to achieve high accuracy when the working load changes. Wen et al. [18] used sparse autoencoder and the maximum mean discrepancy term to transfer training features to testing features; in this way, none of the target fault samples were needed for fault detection. In [19], a new optimal transport-based deep domain adaptation method was presented for rotating machine fault diagnosis. In [20], a deep adversarial domain adaptation (DADA) method was proposed for rolling bearing fault diagnosis; the method builds a DADA network to better address the commonly encountered challenge in real world applications: the distribution of the target domain data is different from that of the source domain data. Note that in typical deep transfer learning [21], it is assumed that the same faults appear in both the training and test stages.

2.2 Zero-shot learning

Recently, Lampert et al. [22] proposed a zero-shot learning (ZSL) scheme, which has received significant attention in the field of image recognition. Instead of using trained objects, it uses a high-level description provided by field experts to detect the target items. The description comprises semantic attributes, e.g., colour, shape and even habits, which could be prelearned without samples of unseen classes. Roughly speaking, ZSL is a method for training models for pattern recognition of unseen types of images based on side information learned from seen classes with relevant description [23]. ZSL has two learning schemes: inductive and transductive models [24]. In the inductive model, only data from seen classes are available during the training stage While in the transductive model, it is assumed that data of both the unseen classes (i.e., unlabeled classes) and seen classes are available for model training, hence it is a type of semi-unsupervised learning. This study is mainly concerned with the inductive scheme.

Zero-shot classification approaches under an inductive setting could be broadly categorized into four groups, namely, direct-attribute prediction based, semantic space embedded based, non-linear multi-modal embedded based, and common space embedded based [25].

For the direct-attribute prediction based method, the most representative model is the direct attribute prediction (DAP) method presented by Lampert et al. [26], which directly builds the relationship between visual features and attributes, and then uses the learned model to predict the attributes of the unseen samples. Lampert et al. [26] also presented an indirect attribute prediction (IAP) method, in which the unseen samples were first assigned to seen classes. Then the unseen samples were predicted using the semantic attribute relationship between seen and unseen types. Note that direct-attribute prediction methods suffer from several drawbacks. Firstly, the two-step prediction method is an indirect approach to find a solution by solving intermediate problems; the solution might be optimal for predicting attributes based on attribute classifiers, but it is not necessarily optimal for predicting classes. Secondly, it is difficult for DAP to extend to incremental learning scenarios. These drawbacks can be overcome by a method based on semantic space embedding discussed below.

The method based on semantic space embedding learns a mapping from features to a semantic space [27, 28]. Frome et al. [29] constructed a deep visual-semantic model by learning a linear mapping from image features to the joint embedding space based on an online learning-to-rank algorithm. Akata et al. [6] presented a label-embedding method for learning a bilinear compatibility function between an image and a label embedding to find the matching embeddings assigned a higher score than the mismatching ones. Akata et al. [30] also proposed a label embedding model for fine-grained classification by combining supervised attributes and unsupervised output embeddings from hierarchies or text corpora. Romera et al. [5] used squared loss as a compatibility function with a regularizer to optimize classification accuracy. Kodirov et al. [31] presented a semantic autoencoder to handle project domain shift problems by reconstructing features after projecting to the semantic space. The approaches based on semantic space embedding enable the representation of visual samples in semantic space and allow recognition in such a space.

A non-linear multi-modal embedded model is usually capable of learning non-linear compatibility relations to optimize projection accuracy or embeddings. Xian et al. [32] extended the bilinear compatibility function to multiple linear (piecewise linear) compatibility functions, making a collection of maps highly interpretable.

Embedding both features and semantic descriptions into a common space is referred to as the common space embedded based method. Changpinyo et al. [33] proposed a synthetic classifier approach for zero-shot classification, which used linear combinations of base classifiers to train classifiers of unseen classes. Hayashi et al. [34] proposed a cluster-based method for multivariate binary classification in ZSL situation, where classifiers (models) were trained based on seen classes first and then used to separate the future data (test data) into two classes: the seen class and unseen (unknown) class. In [35], a novel one-class classification (OCC) method was proposed and used for image classification. The proposed OCC approach can effectively determine whether the input data of interest were from the seen class or the unseen class; the method is potentially very useful for developing and adapting ZSL methods and algorithms.

2.3 Zero-shot learning for fault detection

Zero-shot learning might bring breakthroughs in intelligent fault detection, especially for classifying the types of unseen faults under the condition that samples of these unseen faults are not available for some reason. Many preliminary research results on zero-shot fault detection have already been reported recently in the literature. Lv et al. [36] used a hybrid attribute conditional adversarial denoising autoencoder to tackle the zero-shot fault diagnosis problem. Gao et al. [37] proposed a ZSL method based on contractive stacked autoencoders for bearing fault diagnosis under different working loads. Feng et al. [3] introduced a novel fault description model based on an attribute transfer strategy to classify zero-shot faults in complex mechanical systems. Xing et al. [38] proposed a label description space embedding model for detecting the unseen compound faults of machines. Xu et al. [39] presented a zero-shot intelligent diagnosis method for unseen compound faults of devices using a visual space-based model.

It is worth highlighting that visual attributes (e.g., colour and shape) used for zero-shot image recognition are unsuitable for sensor signal processing (e.g., vibration signals) [3]. When a new type of fault occurs in a system or machine, we will first notice the semantic attribute and description rather than individual samples. For example, from the description “an equipment that converts gas or vapour into liquid and transfers heat from the tube to the air near the tube,” professional workers can detect the object “condenser” without seeing it at all. Similarly, if “high condensing temperature” is a pre-defined fault type, then it is straightforward for us to know that such a fault occurs in the condenser when we are told the high-level attribute information that “high temperature gas from the compressor does not exchange heat well”. Furthermore, it is redundant to design separate attributes for each type of fault because it is not helpful for us to explore that fault attributes defined by humans transcend class boundaries [22], hence the attributes should be shared with different classes of seen or unseen faults. For example, both “reactor cooling water inlet temperature change” and “reactor cooling water valve change” [40] occur at “reactor”, so the attribute “related to reactor” could be shared across the above two seen faults. Then the attribute would be transferred to unseen faults in the testing stage. In conclusion, the fault attributes could include many aspects such as the position of the fault, the related process variable, the size of the fault, etc. The fault attributes provide side information for unseen classes faults, which facilitate the model to detect unseen faults and directly solve the zero-shot fault detection problem.

Feature extraction from raw signals is another considerably crucial process for zero-shot fault detection. Feng et al. [3] used supervised principal component analysis [41] to extract features, under the assumptions that the process control system is linear and follows Gaussian distribution. Such assumptions are strong since most of the data generated by complex industrial processes are non-linear. In [39], 1D vibration signals of interest were transformed to time-frequency images and then fed into a convolutional neural network (CNN) to extract features. It is worth mentioning that converting 1D vibration signals into 2D representations, an additional procedure, is of high computational complexity and needs some application-specific adaptation.

3 Proposed method

3.1 Problem formulation

Following [6], we assume that there is a training (seen) dataset $S = \left\{\left(x_i^s, y_i^s\right)\right\}_{i=1}^{N_s}$ with $x_i^s \in X^s, y_i^s \in Y^s$, which consists of N_s fault data samples and s classes of seen faults. Each sample x_i^s corresponds to a label y_i^s. Likewise, given a testing (unseen) dataset $U = \left\{\left(x_i^u, y_i^u\right)\right\}_{i=1}^{N_u}$ with $x_i^u \in X^u, y_i^u \in Y^u$, the dataset consists of N_u fault data samples and u classes of unseen faults. Each sample x_i^u corresponds to a label y_i^u. The attributes of a fault are denoted as $A = [A^s, A^u] \in R^{L \times C}$, where $L = s + u$, and C is the number of fault attributes. It is important to point out that both A^s and A^u are available in the training stage because the fault attributes are class-level common knowledge rather than expert knowledge. We can obtain the fault attributes in advance. The samples and classes need to meet the following conditions in zero-shot learning settings: $Y^s \cup Y^u = Y$, $Y^s \cap Y^u = \phi$.

3.2 Model structure

The proposed method, SSB-ZSL-1DCNN, motivated by and adapted from the idea of semantic space embedding, comprises three steps: feature extraction, human-defined label embeddings, and a feature embedding model. The overall structure of the method is presented in Fig. 1.

1) *Feature extraction*: 1D CNN is preferable when tackling industrial fault 1D signals since 1D CNN is easier to train and needs lower computational complexity than 2D convolutions [42]. Therefore, we use 1D CNN as the feature extractor. The architecture of the designed 1D CNN is shown in Table 1. It contains two convolution layers, two max-pooling layers, one flatten layer and one fully-connected layer. The inputs of the 1D CNN are the 1D time-series signals, and the outputs from the fully-connected layer are extracted features.

Table 1 The architecture of the designed 1D CNN for feature extraction

Layer Name	Description	Kernel Size	Stride	Kernel Number
C1	Convolution	1×3	3×1	128
P1	Max-pooling	1×4	/	128
C2	Convolution	1×3	3×1	128
P2	Max-pooling	1×4	/	128
F	Flatten	1×128	/	1
FC	Fully-connected	1×64	/	1

2) *Human-defined label embeddings*: In practical fault diagnosis, tagging each fault sample is complex and time-consuming. Fault attributes (represented by a matrix A in the Section 4) provide side information which can be used to establish the relationships between seen faults and unseen faults. Fault attributes allow for sharing characteristics of faults such as fault position and fault effect, which are easily annotated by experts and transformed into computer-readable vector forms [30]. The description of each attribute could be a binary value $\phi^{0,1} \in \{0, 1\}$ or a continuous value $\phi^C \in [0, 1]$ for each class. The attributes for each fault class can be written as:

$$\Phi(y) = \left[\phi_{y,1}, \ldots, \phi_{y,E}\right]^T \tag{1}$$

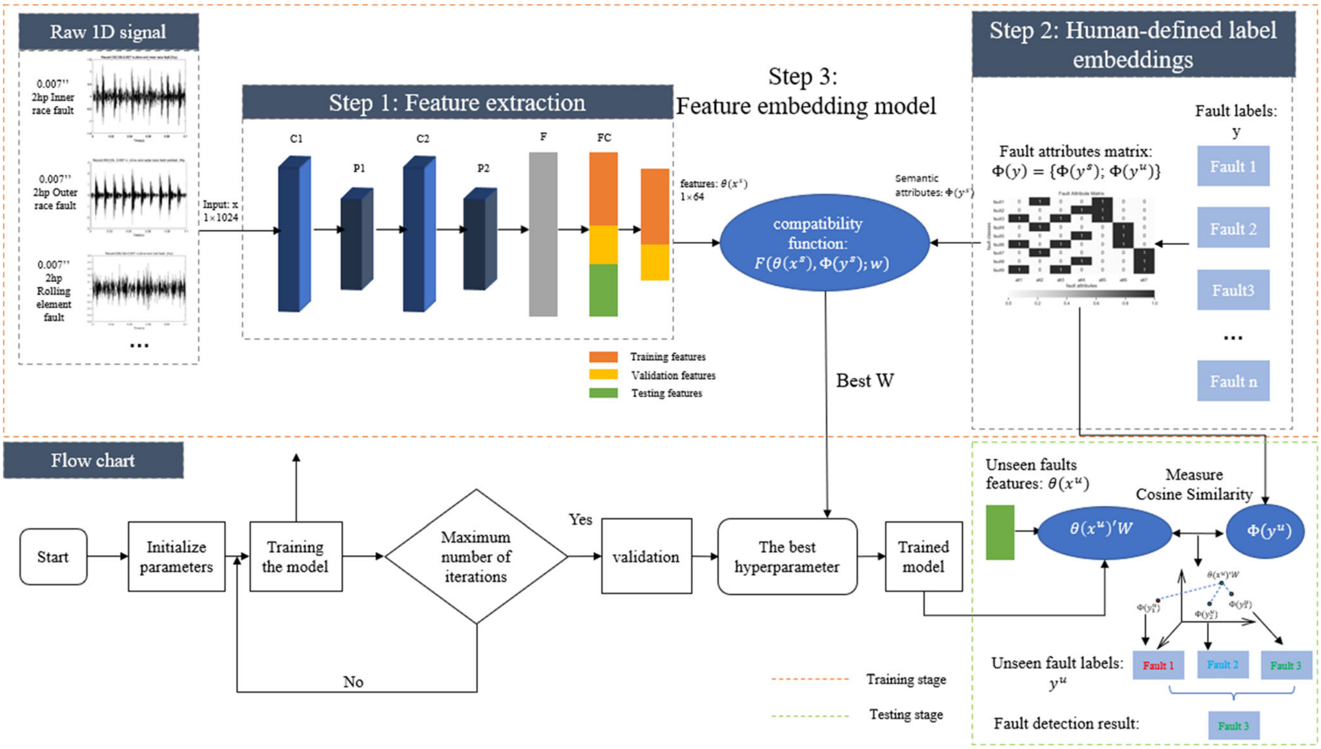

Raw 1D signal
0.007'' 2hp Inner race fault
0.007'' 2hp Outer race fault
0.007'' 2hp Rolling element fault
...
Input: x 1×1024
Step 1: Feature extraction
C1
P1
C2
P2
F
FC
Training features
Validation features
Testing features
features: $\theta(x^s)$ 1×64
Step 3: Feature embedding model
compatibility function: $F(\theta(x^s), \Phi(y^s); w)$
Semantic attributes: $\Phi(y^s)$
Best W
Step 2: Human-defined label embeddings
Fault attributes matrix: $\Phi(y) = \{\Phi(y^s); \Phi(y^u)\}$
Fault labels: y
Fault 1
Fault 2
Fault3
...
Fault n
Flow chart
Start
Initialize parameters
Training the model
Maximum number of iterations
Yes
No
validation
The best hyperparameter
Trained model
Training stage
Testing stage
Unseen faults features: $\theta(x^u)$
$\theta(x^u)'W$
Measure Cosine Similarity
$\Phi(y^u)$
$\Phi(y_1^u)$
$\Phi(y_2^u)$
$\Phi(y_3^u)$
Unseen fault labels: y^u
Fault 1
Fault 2
Fault 3
Fault detection result:
Fault 3

Fig. 1 The structure of the proposed method

where $\phi_{y,1}$ could be one of the binary numbers in $\{0, 1\}$ or a real number between 0 and 1; y denotes fault class, and E denotes the dimension of attributes for a fault class. Note that continuous attributes ϕ^C carry more information than binary attributes $\phi^{0,1}$. For illustration purposes here, we consider the attribute matrix in binary, but in the subsequent experiments, we set random and continuous attributes rather than binary attributes. If a fault does not have this attribute, set it to 0; if it has this attribute, set it to a random number in the range of (0,1).

3) *Feature embedding model*: We define a prediction function f by maximizing the bi-linear compatibility function F as follows:

$$f(x; w) = \arg \max_{\mathrm{y} \in \mathrm{Y}} \mathrm{F}(\mathrm{x}, \mathrm{y}; \mathrm{w}) \tag{2}$$

where w denotes the parameter vector of F and can be written as a $D \times E$ matrix W that D is the extracted features dimension and E is the attributes dimension. The bi-linear form of the compatibility function $F : X \times Y \rightarrow R$ between a raw fault data space X and a fault label space Y can be defined as follows:

$$F(x, y; W) = \theta(x)^T W \Phi(y) \tag{3}$$

where the extracted features are denoted by $\theta(x)$ and fault label embedding is denoted by $\Phi(y)$. $F(x, y; W)$ is an optimized compatibility function based on the ranking, enabling that the correct label will get the highest rank than any other labels by learning W. This idea is closely related to the web scale annotation by image embedding (WSABIE) algorithm [43] which learns a low-dimensional joint embedding space for both images and annotations to classify the annotations from the ranked list of annotations. The significant difference between our method and WSABIE is that the latter learns both $\Phi(y)$ and W, whereas the former only learns W and uses fault attributes as side information $\Phi(y)$.

4) *Parameterestimation*: similar to the formulation defined in the unregularized structured SVM [43], the weighted approximate ranking objective function is to minimize:

$$\sum\nolimits_{y \in Y} \frac{\beta_{r_{\Delta(x_n, y_n)}}}{r_{\Delta(x_n, y_n)}} \sum\nolimits_{y \in Y} \max\{0, \mathrm{s}(\mathrm{x_n}, \mathrm{y_n}, \mathrm{y})\} \tag{4}$$

where

$$s(x_n, y_n, y) = \Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W) \tag{5}$$

$$\beta_k = \sum\nolimits_{i=1}^{k} \alpha_i \tag{6}$$

$$r_{\Delta(x_n, y_n)} = \sum\nolimits_{y \in Y} 1(\mathrm{s}(\mathrm{x_n}, \mathrm{y_n}, \mathrm{y}) > 0) \tag{7}$$

Here, $s(x_n, y_n, y)$ is misclassification loss function with margin $\Delta(y_n, y)$, where $\Delta(y_n, y) = 1$ if $y \neq y_n$ and 0 otherwise. As suggested in the WSABIE algorithm, we choose $\alpha_i = 1/i$. $r_{\Delta(x_n, y_n)}$ is the upper bound on the rank of fault label y_n related to fault data x_n. The ranking-based function (4) aims to obtain higher compatibility between the feature extraction and fault label embedding of the target fault label than between the feature extraction and fault label embedding of the wrong fault labels.

In the training stage, we use the extracted features $\theta(x)$ and fault attributes $\Phi(y)$, which are only from seen fault classes, to learn W. We apply stochastic gradient descent (SGD) to optimize W and then find the highest scored class y, if arx $_{y \in Y}$max $s(x_n, y_n, y) \neq y_n$:

$$W^{(t)} = W^{(t-1)} + \eta_t \beta_{\lfloor \frac{N-1}{k} \rfloor} \theta(x_n) [\Phi(y_n) - \Phi(y)]^T \tag{8}$$

where η_t is the learning rate at iteration t; in this study, a constant step size $\eta_t = \eta$ is used. Based on WSABIE, $r_{\Delta(x_n, y_n)}$ is approximated as $r_{\Delta(x_n, y_n)} \approx \lfloor \frac{N-1}{k} \rfloor$, where N is the number of fault labels and k is the number of wrong fault labels. After the completion of the training stage, the best W can be obtained.

In the testing stage, we embed a feature onto the best W and use the cosine similarity measure to search for the nearest fault attribute vector, which belongs to one of the unseen fault classes.

4 Experiments

This section presents two case studies for two real datasets: rolling bearing fault dataset created by Case Western Reserve University (CWRU) and chemical process control fault dataset known as Tennessee-Eastman process (TEP). The two case studies were carried out from different perspectives to comprehensively evaluate the performance of the proposed method.

4.1 The Case Western Reserve University (CWRU) dataset

4.1.1 Introduction to the CWRU dataset

The CWRU dataset, consisting of vibration-based rolling bearing fault data, is from the Case Western Reserve University Bearing Data Center [44]. The test bench, shown in Fig. 2, contains a 2 hp reliance electric motor, a torque and a dynamometer. In addition, an acceleration sensor is installed above the bearing housing at the fan-end and drive-end to collect the vibration acceleration when collecting the fault data.

The faults are located on drive-end bearing and fan-end bearing, respectively, which contain inner race fault, rolling element fault and outer race fault with four working loads (0,1,2 and 3 hp). The fault dimeter for each type of faults ranges from 0.007 to 0.028 inch on the bearings using electro-discharge machining (EDM). As for the variables in each class of faults, there are drive-end acceleration data (DE), fan-end acceleration data (FE), base plate acceleration data (BA) and motor speed (RPM). The sampling rate of signals for the dataset is 12 kHz.

We chose to use the 12 kHz drive-end bearing fault data as our experimental dataset. Overall, there are four groups of experiments, namely, 0 hp, 1 hp, 2 hp and 3 hp. For each group, there are nine kinds of faults in total, and only DE is selected as a variable because the vibration signal, collected at the drive-end, is more comprehensive and less infected by other components and environmental noise. For graphical illustration purposes, the vibration signal samples of the rolling race fault with 3 hp and 0.021 inch fault diameter are shown in Fig. 3, where the top panel shows the waveform of the signal in the time domain and the bottom panel shows the corresponding spectrum. For each type of fault, the first 102,400 data for DE are considered; the data were then pre-processed and rearranged with an overlap sampling approach, with the overlap ratio of 50%, resulting in a total of 200 samples, each consists of 1024 data points. The details of faults in the dataset are given in Table 2.

Note that different sizes (diameters) of failures can damage the equipment to different degrees, some operating conditions do not allow for larger size failures, and few factories will be allowed to run to large size failures and collect samples for training. Therefore, the largest size failures might have zero sample for model training, it is extremely difficult (if not impossible) for traditional multi-classification methods to detect unseen faults. As a result, the proposed zero-shot fault detection method is meaningful and realistic.

4.1.2 Model implementation

The first stage of this method is feature extraction. The designed 1D CNN was applied to extract fault features from raw data. As shown in Table 1, the architecture of the 1D-CNN model for vibration signals is constructed with two Conv layers and one FC layer. The input signal of the 1D-CNN is of size 1 × 1024 and the output of the FC layer is of size 1 × 64.

To illustrate the feature extraction results intuitively, the t-SNE (t-distributed stochastic neighbor embedding) algorithm was employed to provide a 2D representation of the features as the output of the FC layer. Taking the case of 1 hp working load as an example, the distributions and clusters of the nine types of faults after t-SNE are presented in Fig. 4, where the horizontal and vertical coordinates represent the first two principal components extracted from t-SNE, respectively.

The second stage is to build human-defined fault label embeddings for the preprocessed experimental dataset, i.e., the fault attribute matrix $\boldsymbol{A}$, which is shown in Fig. 5, and the meaning of each attribute is displayed in Table 3. For each bearing fault, seven fine-grained fault attributes were specified by using the statements given in Table 2. Note that the attribute matrix does not distinguish the same type of faults under

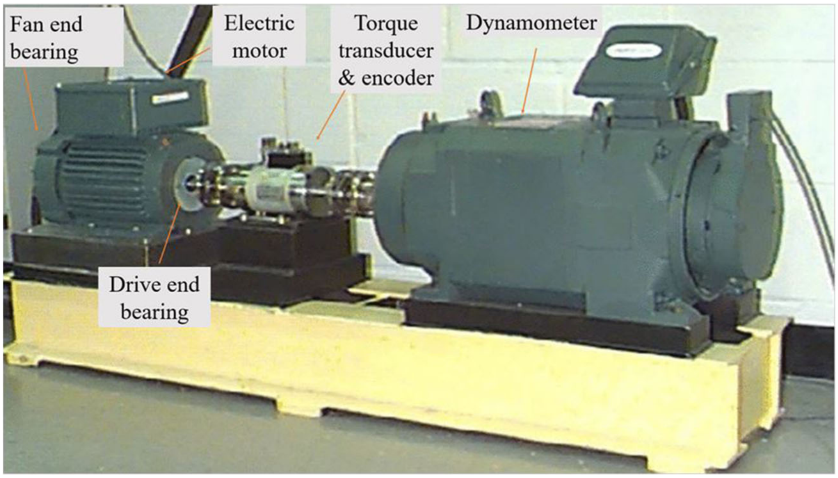


Fig. 2 The CWRU test bench

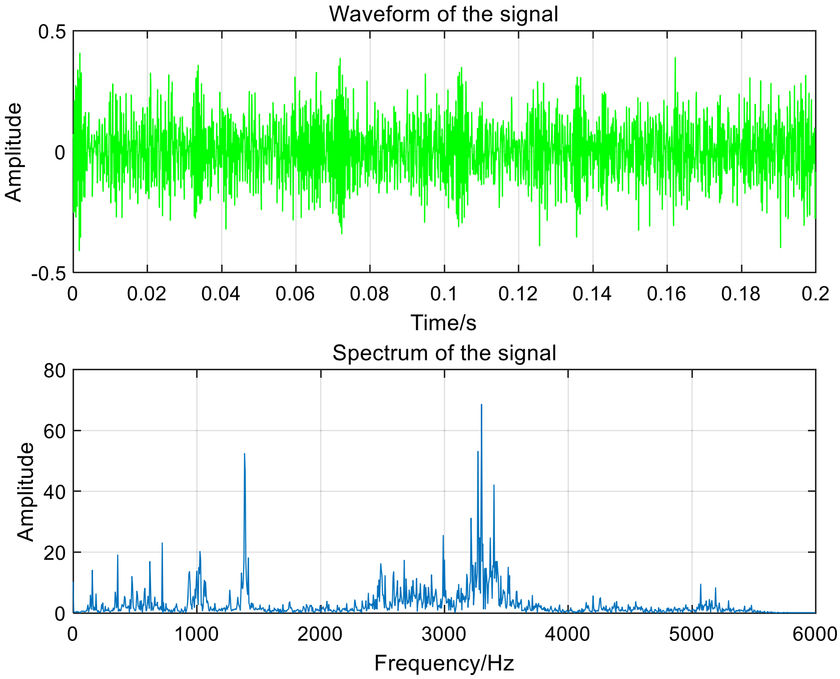


Fig. 3 The 3 hp 0.021'' rolling element fault signal (time domain) and its spectrum (frequency domain)

different working loads. For example, the two types of faults 1 hp 0.007'' inner race fault and 2 hp 0.007'' inner race fault have the same attributes. Once fault attributes are obtained in the preparation and training stage, the resulting classifier can then detect unseen faults without using samples of the unseen faults in the training stage.

Finally, we put the extracted features into the feature embedding model to match the most similar fault attributes to find the corresponding unseen fault categories. The ordinary cosine distance was used to measure the similarity between attributes.

There are a total of 36 types of bearing faults in total, divided into four groups of experiments; each group has nine types of faults. The dataset was split as follows: four types of 0.007'' (inch) and 0.014'' faults were used for training the models, two types of 0.007'' and 0.014'' faults were used for validation, and the rest three types of 0.021'' faults belong to the test set. The data split for the train/validation/test sets is displayed in Table 4. The numbers of training, validation and test samples are 4 × 200 = 800, 2 × 200 = 400, and 3 × 200 = 600, respectively.

Table 2 Fault labels (9 features in each working load)

Fault Number	Fault Description
1	0.007'' inner race fault (0–3 hp)
2	0.007'' rolling element fault (0–3 hp)
3	0.007'' outer race fault (0–3 hp)
4	0.014'' inner race fault (0–3 hp)
5	0.014'' rolling element fault (0–3 hp)
6	0.014'' outer race fault (0–3 hp)
7	0.021'' inner race fault (0–3 hp)
8	0.021'' rolling element fault (0–3 hp)
9	0.021'' outer race fault (0–3 hp)

As for performance evaluation, we are interested in the accuracy of each type of unseen fault, so the average per-class top-1 defined below is used to measure the accuracy [28]:

$$acc = \frac{1}{\|Y^u\|}\sum_{y^u}^{\|Y^u\|}\frac{\#correct\ detections\ in\ y^u}{\#samples\ in\ y^u} \tag{9}$$

where y^u is an unseen fault of large size and Y^u is the number of the unseen faults.

4.1.3 Accuracy of zero-shot fault detection

The results of zero-shot fault detection are presented in Table 5. We used six types of 0.007'' and 0.014'' bearing faults to train the models, and then used the resulting classification models to detect three types of 0.021'' bearing faults. The accuracies vary from 66.67 to 87.67%, under different working loads from 0 hp to 3 hp. Clearly, the performance of the proposed method is significantly better than the chance level of 33.33%, which indicates that it is possible to detect unseen large size fault without using their samples in the training stage, and this was achieved by sharing fault attributes with seen small-size faults and unseen large-size faults.

As for the individual result of each group, the proposed method performs better for the groups of 0 hp and 2 hp working loads than for 1 hp and 3 hp working loads. We analyzed

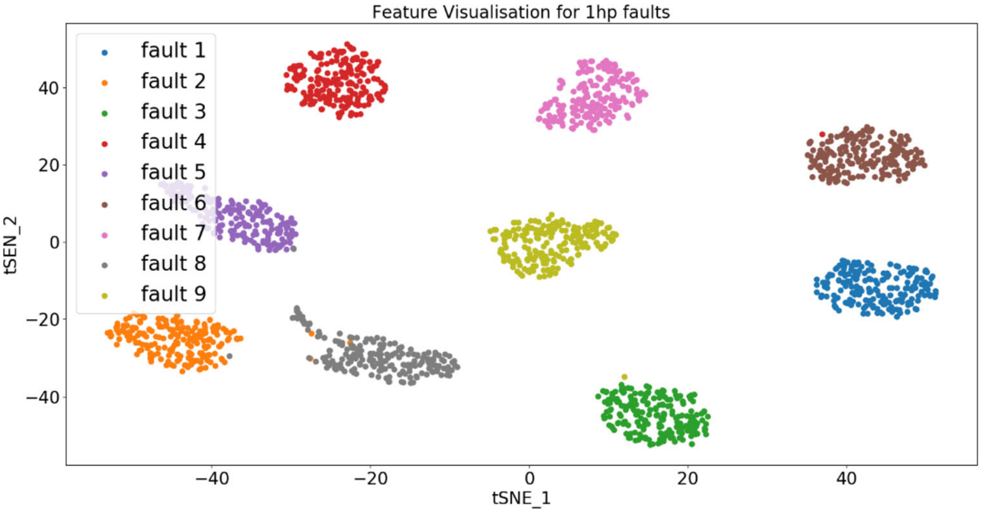
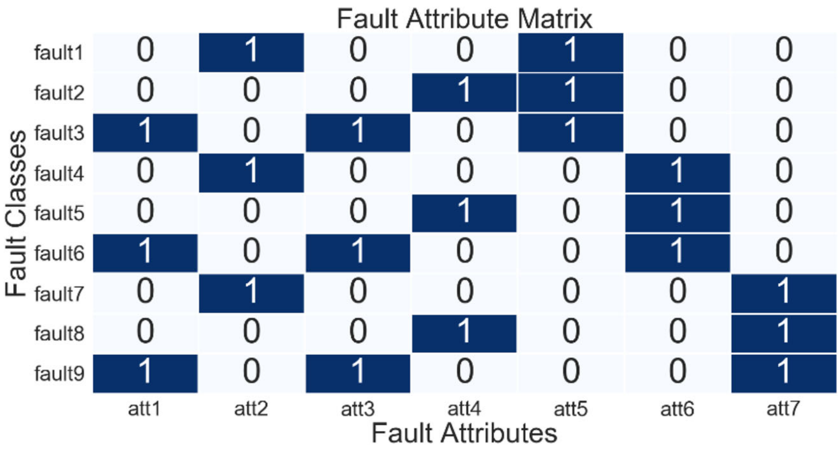


Fig. 4 The t-SNE results of the output features of the FC layer for the case of working load 1 hp. The horizontal and vertical coordinates represent the first two principal components extracted from t-SNE, respectively


Fault Attribute Matrix
Fault Classes
Fault Attributes

	att1	att2	att3	att4	att5	att6	att7
fault1	0	1	0	0	1	0	0
fault2	0	0	0	1	1	0	0
fault3	1	0	1	0	1	0	0
fault4	0	1	0	0	0	1	0
fault5	0	0	0	1	0	1	0
fault6	1	0	1	0	0	1	0
fault7	0	1	0	0	0	0	1
fault8	0	0	0	1	0	0	1
fault9	1	0	1	0	0	0	1

Fig. 5 Fault attribute matrix *A*

the confusion matrices for these two groups, which show that fault 9 (0.021'' outer race fault) has a chance of being misclassified to fault 7 (0.021'' inner race fault). To investigate the reason of the misclassification, we further analyzed the time-domain signals of fault 7 and fault 9 in 3 hp and their frequency-domain properties (spectra) which are shown in Fig. 6, from which it can be observed that the two signals nearly have the same resonance band at around 3000 Hz in the frequency domain. Therefore, inner race fault and outer race fault are similar in the frequency domain, implying that it is more difficult to distinguish these two types of faults than the rolling element faults. Nevertheless, in the following experiments we will demonstrate that our results are pretty competitive compared with many state-of-the-art methods.

Table 3 Side Information of the Attributes for the CWRU dataset

Attribute Number	Fault Attributes
att1	Centered position
att2	Occurred at inner ring
att3	Occurred at outer ring
att4	Occurred at rolling element
att5	Fault diameter 0.007''
att6	Fault diameter 0.014''
att7	Fault diameter 0.021''

Table 4 The Four groups of data split for the CWRU dataset

Group.	Training Faults	Validation Faults	Test Faults
0 hp	1, 2, 4 ,6	3, 5	7, 8, 9
1 hp	1, 2, 4, 6	3, 5	7, 8, 9
2 hp	1, 2, 4, 6	3, 5	7, 8, 9
3 hp	1, 2, 4, 6	3, 5	7, 8, 9

Table 5 The zero-shot fault detection results for the CWRU dataset (%)

Group.	Step Size	Accuracies
0 hp	4	78.50
1 hp	0.01	66.67
2 hp	0.01	87.67
3 hp	4	66.78

Table 6 Performance comparison of different zero-shot learning method for fault detection of the CWRU dataset (%)

Group.	0 hp	1 hp	2 hp	3 hp
SJE	38.17	39.17	32.17	39.50
DEVISE	39.67	38.83	29.33	40.50
SAE	24.17	31.50	26.67	33.17
ESZSL	56.50	28.30	60.00	27.76
Reference [3]	51.34	59.59	54.34	61.42
SSB-ZSL-1DCNN	78.50	66.67	87.67	66.78

4.1.4 Comparison with other ZSL methods

We compare five state-of-the-art zero-shot learning methods mentioned in Section 1 (Introduction), namely, SJE [30], DEVISE [29], SAE [31], ESZSL [5], and the zero-shot fault detection method from [3], under the same setting. It should be noticed that these methods were designed for image classification, where image attributes were defined and used for discrimination purposes. However, the visual attributes proposed in these methods are not applicable to time series signal-based fault detection tasks. Hence, in this study, we incorporate the newly designed fault attribute matrix (shown in Fig. 5) to our proposed SSB-ZSL-1DCNN model, and use it to replace the image attributes and label information employed in these compared methods. For a fair comparison, we train SJE, DEVISE, SAE and ESZSL models using the same raw data. The results produced by the trained models are presented in Table 6.

From Table 6, it is clear that our method significantly outperforms the five compared methods: the first four are semantic space embedded ZSL methods and the fifth is a fault description-based attribute transfer approach for ZSL. The main reasons that these compared methods show very low classification accuracies may be explained as follows. These methods which were initially designed for 2D data (images) could work well for image recognition and classification tasks based on the extracted features. However, for 1D data considered in this study, these methods may experience considerable degradation since they could not find useful image features from the given 1D vibration signals, and the required visual attributes are not available, either. All this suggests that the exploration and use of good feature extraction methods which can effectively find most useful and representative features for 1D time-series fault data are highly needed. Hence, in the following, we apply two groups of feature extraction methods to the experimental dataset and evaluate their accuracy performances: deep learning based methods (VGG16, VGG19 and Resnet50) and traditional methods (PCA, ICA and KPCA).

4.1.5 Performance comparison of feature extraction methods

In this section, we evaluate the performance of the proposed method for feature extraction and compare it with other two groups of feature extraction methods: three deep learning-based methods and three traditional feature extraction methods.

Most CNNs are designed for learning from 2D data (e.g. images). However, in the field of fault detection and diagnosis of industrial systems, signals are represented as 1D time series in most cases. Typical CNN models cannot be directly applied to such tasks. A way of using CNNs to handle 1D signals is to convert 1D signals to 2D data. In this study, we use

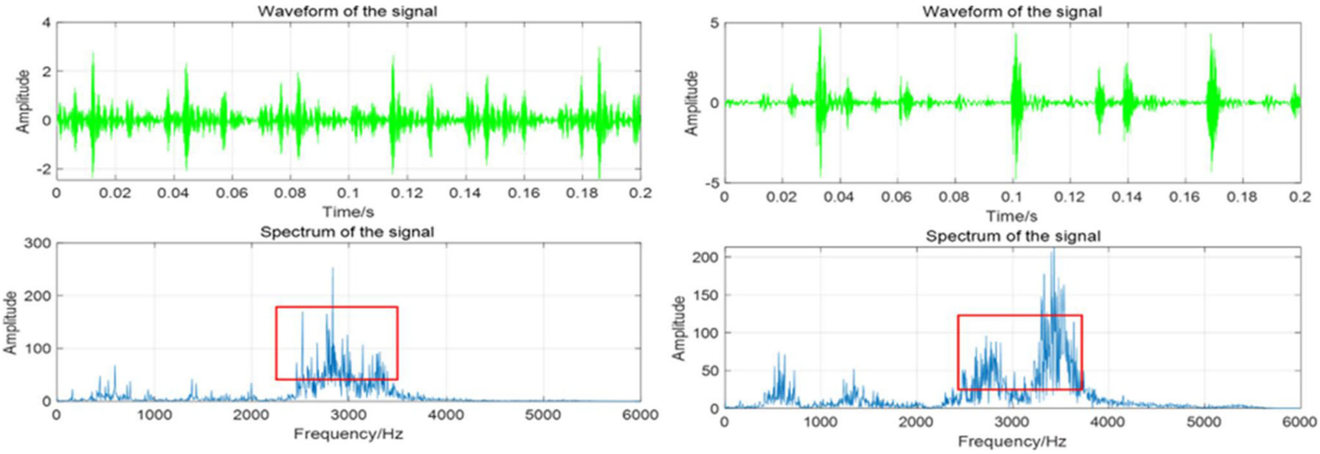


Fig. 6 Time-domain signal and frequency-domain signal of fault 7 and fault 9 in 3hp

continuous wavelet transform to obtain time-frequency images. As a result, each output image is a 3-channel RGB image of 236 × 236 × 3. In order to avoid the effect of the network colour bars, the RGB image is transformed into a 224 × 224 time-frequency grayscale image. Next, each type of fault images are fed into the following three 2D deep neural network methods: VGG16, VGG19 [45] and Resnet50 [46], all of which were pre-trained on ImageNet 1 K classes to determine the initial values of the parameters. The detailed information about these three models is shown in Table 7. The implementation procedures of the three methods are depicted in Fig. 7.

For the three traditional feature extraction methods, we firstly perform an overlap sampling procedure to the raw signals, to generate input data for PCA, ICA and KPCA. The output of each of the feature extraction methods is a 64 × 1 dimensional vector, whose length is the same as that of the 1D CNN proposed in this study.

The performances of six compared methods and the proposed method are shown in Table 8, from which it can be seen that the 1D CNN model designed in this study performs far better than the other three deep CNN methods. These results strongly confirm that 1D CNN is very efficient and promising for solving fault detection of industrial systems. It has the following advantages: (1) 1D CNN has much lower computational complexity than 2D CNN; (2) 1D CNN has fewer hidden layers and simpler architecture than 2D CNN, which means that less time is needed to train and implement; (3) Unlike 2D CNNs which need GPU or high performance computational resources, 1D CNN is feasible to implemted and operate in normal CPU, this significantly reduces cost in many real applications.

Meanwhile, compared with the three traditional feature extraction methods, the propsoed SSB-ZSL-1DCNN model also shows significantly much better performance, especially when compared with the two linear feature extraction methods, PCA and ICA. For the case of ‘3 hp’, our proposed method, SSB-ZSL-1DCNN, achieves an accuracy performance of 66.78%, which is slightly better than that of KPCA. However, for other cases, SSB-ZSL-1DCNN peforms far better than the best results of the three basline methods, with 18.67%, 11.00% and 27.17% increase in accuracy, respectively.

In addition, when compared with the attribute transfer method [3], our model performs better for all the cases, especially for the two gropus of 0 hp and 2 hp. There are several reasons that may explain the better performance of the propsoed method. Firstly, SSB-ZSL-1DCNN is an end-to-end model. The structure of the model used in the baseline method is not in an end-to-end manner, as it just trains an attribute learner for each attribute, and classify the faults based on the outputs of attribute learners, making the classification results rely heavily on the accuracy of the attribute learners. Secondly, the classifiers used in the compared method are LSVM (linear support vector machine), RF (random forest) and NB (naïve Bayes), of which LSVM is a linear approach. The results from LSVM are much worse than from the other two. In the comparisons, we used the average accuracy of the three classifiers. Thirdly, the feature extraction method used in the baseline method is supervised PCA, which is a linear feature reduction method, requiring the training samples to follow a Gaussian distribution; such a requirment, however, may not be met for the CWRU dataset.

Regarding computational cost, the two groups of feature extaction methods need completely different running times, as shown in in Table 9. The average running time of the three traditional methods is 1.83s, whereas the average running time of VGG16, VGG19 and Resnet50 is 2485.06s. The overall time used by proposed SSB-ZSL-1DCNN method is 71.98s for 20 epochs. Clearly, the deep learning based methods need far more time for training, and usually the deeper the CNN, the more time it needs. Compared to other convolutional networks and traditional methods, the propsoed 1D CNN model only needs a relatively small amount of time and shows obviously, significantly better results. Putting running time and classification accuarcy together, SSB-ZSL-1DCNN has excellent adaptability to 1D time series signals and shows high efficiency.

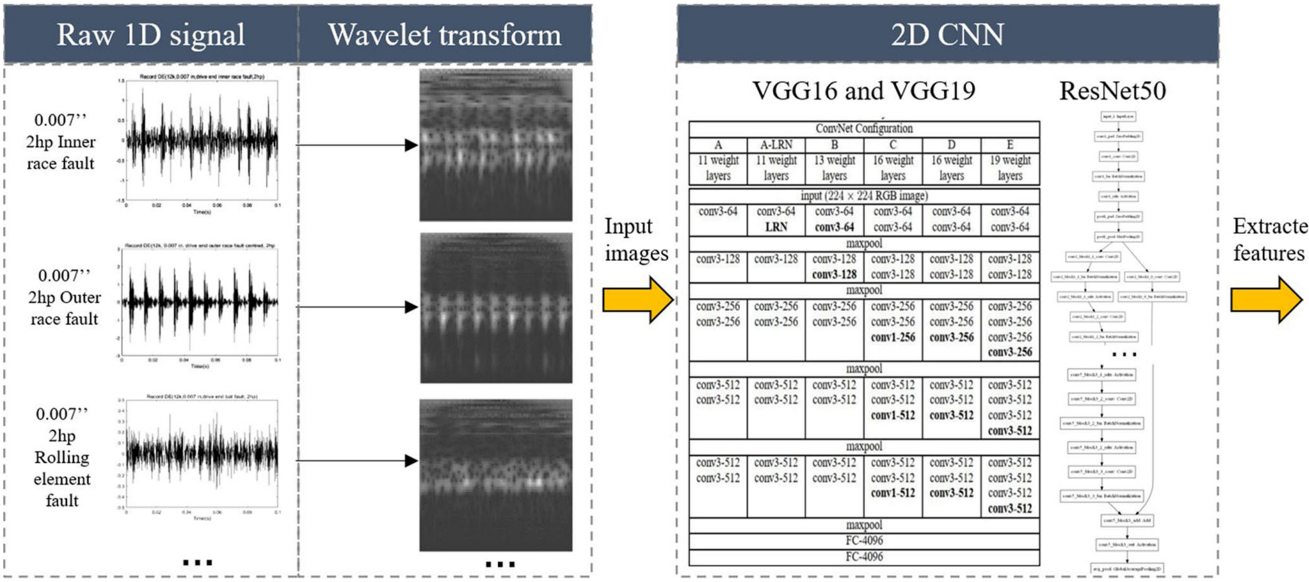
4.2 The Tennessee-Eastman Process (TEP) dataset

4.2.1 Introduction to the TEP dataset

The Tennessee-Eastman dataset [39] was collected for a comprehensive industrial chemical process by Eastman Chemical Company. This dataset is widely studied in the field of fault detection. The process contains five major operations, including a reactor, a product condenser, a vapor-liquid separator, a recycle compressor and a product stripper. Since this plant-wide industrial process has many kinds of faults, it could be challenging to collect enough samples to build a fault

Table 7 Experimental settings for deep learning-based feature extraction methods

Feature Extraction Method	Input Shape	Output Layer	Output Shape
VGG16	224×224	The second fully-connected layer	4096×1
VGG19	224×224	The second fully-connected layer	4096×1
Resnet50	224×224	The last global average pooling layer	2048×1



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					

Fig. 7 Procedure of 2D feature extraction

detection system and might bring considerable losses to the factory; a zero-shot fault detection method therfore becomes necessary to detect specific faults for such a system without their samples in the training stage.

A total of 15 types of faults are exploited in this case study, and the details are introduced in Table 10. Each type of fault contains 480 samples, and each sample invovles 52 variables (fetures) in total.

Table 8 Performance comparison of different fault feature extraction methods for the CWRU dataset (%)

Group.	VGG16	VGG19	Resnet50	PCA	ICA	KPCA	1D CNN
0 hp	32.83	40.17	46.33	35.00	35.83	59.83	78.50
1 hp	33.17	40.17	33.33	36.83	38.67	55.67	66.67
2 hp	50.83	51.50	61.67	29.00	30.33	60.50	87.67
3 hp	51.00	63.17	51.33	36.50	37.00	62.00	66.78

Table 9 Computational time of different feature extraction methods for the CWRU dataset

Feature Extraction Method	Computational time (s)
PCA	1.435524
ICA	2.174596
KPCA	1.873985
VGG16	1483.859375
VGG19	1522.639438
Resnet50	4448.684739
SSB-ZSL-1DCNN	71.978182

4.2.2 Model implementation

The structure of the 1D CNN is the same as that for CWRU described in Section 4.1.2. As for the fault attribute matrix in the second step, we use the fault attribute matrix ***A*** proposed in [3], which is shown in Table 11; Fig. 8. There are a total of 20 human-defined fault attributes shared between seen faults and unseen faults.

Table 10 Fault description for the TEP dataset

Fault Number	Fault Description	Type
1	A/C feed ratio, B composition constant	Step
2	B composition, A/C ratio constant	Step
3	D feed temperature	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss	Step
7	C header pressure loss	Step
8	A, B,C feed composition	Random variation
9	D feed temperature	Random variation
10	C feed temperature	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variable
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking

Table 11 Fault attributes for the TEP dataset

Number	Fault Attribute
Att1	Input A is changed
Att2	Input C is changed
Att3	A/C ratio is changed
Att4	Input B is changed
Att5	Related with pipe4
Att6	Temperature of input D is changed
Att7	Related with pipe2
Att8	Disturbance is step-changing
Att9	Input is changed
Att10	Temperature of input is changed
Att11	Occurred at reator
Att12	Temperature of cooling water is changed
Att13	Occurred at condenser
Att14	Related with pipe1
Att15	Disturbance is ramdom varying
Att16	Model parameters are changed
Att17	Disturbance is slow drift
Att18	Related with cooling water
Att19	Related with valve
Att20	Disturbance is sticking

As for the experimental setting, we split the 15 types of faults into three parts: eight for training (seen), four for validation and three for test (unseen) faults. To comprehensively evaluate the proposed method, we follow the practice in [3] to divide the TEP data into five groups, the details are shown in Table 12. The number of training samples is 5760, and the test samples is 1440. The evaluation criteria are the same as that of CWRU dataset.

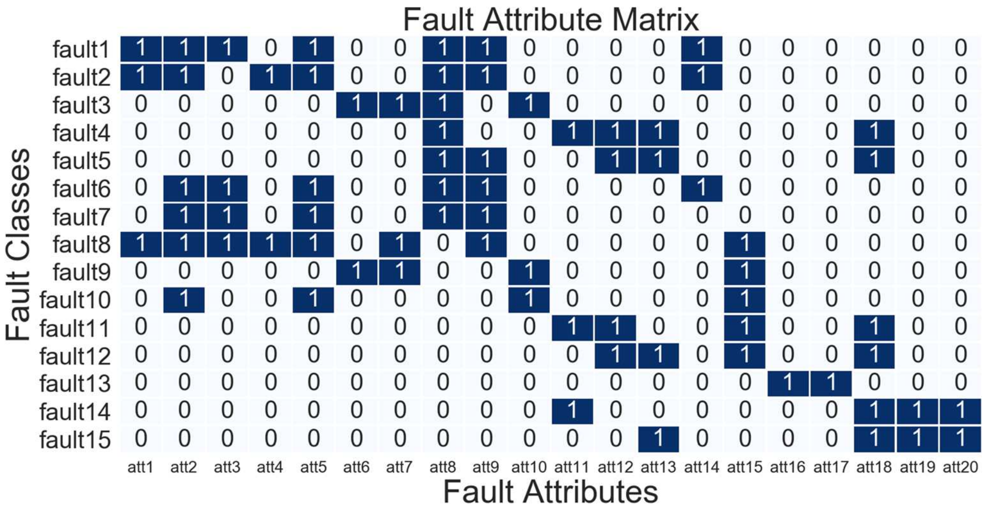
Table 12 Five groups of sub-datasets for the TEP dataset [3]

Group.	Training Faults	Validation Faulrs	Test Faults
A	3, 5, 7, 9–13	2, 4, 8, 15	1, 6, 14
B	1, 3, 5, 6, 8, 12, 13, 15	2, 9, 11, 14	4, 7, 10
C	1, 2, 4, 5, 7, 8, 10–12, 14, 15	3, 6, 9, 13	8, 11, 12
D	1, 6, 7, 9–11, 13, 14	4, 8, 12, 15	2, 3, 5
E	1, 2, 4–6, 8–11	3, 7, 12, 14	9, 13, 15

4.2.3 Fault detection accuracy with ZSL

The fault detection results of ZSL for the 5 groups with of the TEP data are shown in Table 13. The detection accuracies vary from 59.72 to 96.67% for different types of unseen faults. The accuracy details of each unseen fault are presented in the confusion matrices in Fig. 9. For some specific classes, such as fault 8 in Group C and fault 2 in Group D, the accuracies are 39% and 33%, respectively, a sort of chance level of 33.33%, but none is below that the chance level. From the fault description in Table 9, we can see that fault 2 and fault 8 are two complicated faults, involving three-quarters of the components in the chemical reactions. Hence, given the nature and mechanism of ZSL method, it is more difficult to train an accurate model based only on the seen fault data with few fault attributes, to detect the unseen faults which invovlve much more fault attributes.

From Table 13, the proposed method shows a much better performance in Group A, B, C and E, although its accruacy is slightly below that of [3] in Group D. Note that for Group E, the accuracy of our method is far higher than that reported in [3]. To explain such a large differenc in accruacy, we checked the confusion matrices presented in [3], the accuracy was 26% for fault 5 and 21% for fault 9, which are below the random



	att1	att2	att3	att4	att5	att6	att7	att8	att9	att10	att11	att12	att13	att14	att15	att16	att17	att18	att19	att20
fault1	1	1	1	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0
fault2	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0
fault3	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
fault4	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0
fault5	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	0	0
fault6	0	1	1	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0
fault7	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
fault8	1	1	1	1	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
fault9	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0
fault10	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
fault11	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0
fault12	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0
fault13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
fault14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1
fault15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1

Fig. 8 Fault attribute matrix A [3]

Table 13 Fault detection results for the TEP dataset with the proposed ZSL method (%)

Group.	Step Size	SSB-ZSL-1DCNN	Reference [3]
A	0.7	84.31	75.78
B	0.5	76.94	57.54
C	0.8	62.08	53.46
D	0.7	59.72	62.50
E	0.7	96.67	51.03

choice chance level (33.33%). The accuracies of our mehtods for the three types of faults, however, reach 98%, 93% and 99%, respectively.

4.2.4 Comparison with traditional feature extraction methods

In the previous section, we have compared different deep learning based feature extraction methods for the CWRU dataset. We now focus on three commonly used traditional feature extraction methods, namely, PCA, ICA and KPCA, to further evaluate and compare the performance of the propsoed method.

For the three feature extraction methods, PCA, ICA and KPCA, we implemented the experiments by using the scikit-learn package [47]. As for the parameter settings, each method adopts 20 features extracted from a total 52 raw variables. The accuracies of different feature extraction methods are presented in Table 14. For all groups (but Group D), we achieve impressive results. For Group D, the proposed SSB-ZSL-1DCNN model achieved a slightly lower accuracy than ICA. However, we obtained significant improvements for Groups A-C: 15.84%, 20.06% and 16.25% higher compared with the best results of the other three methods, respectively. In Table 15, we also compare the computational time used by PCA, ICA, KPCA and SSB-ZSL-1DCNN. The time used by KPCA is more than twice of that used by 1D CNN. Althogh the running time of PCA and ICA is shorter than SSB-ZSL-1DCNN, their overall accuracy perfromances are obvously much lower than our method.

From the above comparisons, it can be concluded that the proposed method performs better than the traditional feature extraction methods. The SSB-ZSL-1DCNN model works well for 1D time-series fault signals and can

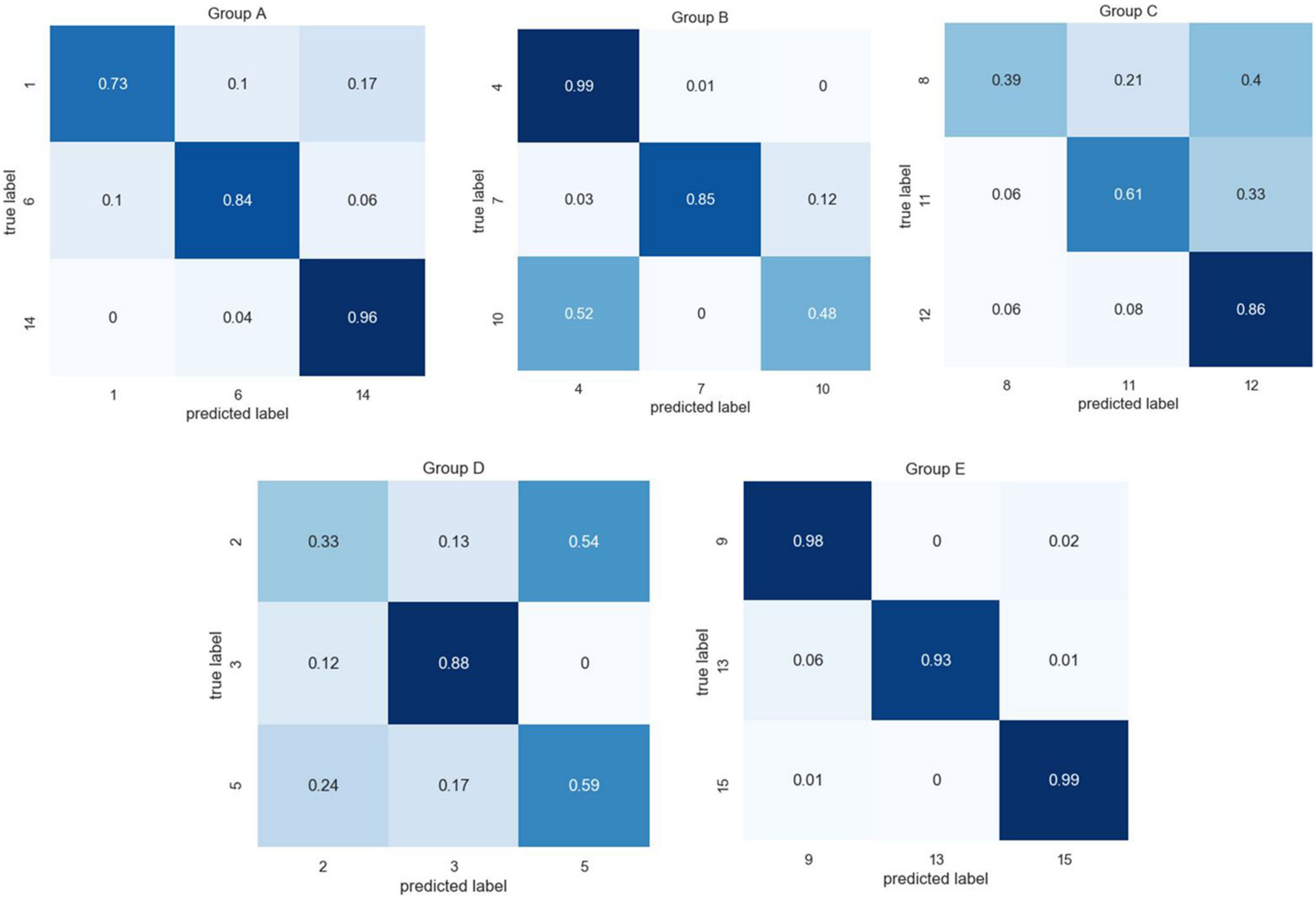


Fig. 9 Confusion matrices of the results of unseen faults

Table 14 Performance comparison of different feature extraction methods for the TEP dataset (%)

Group	PCA	ICA	KPCA	SSB-ZSL-1DCNN
A	61.39	68.47	63.54	84.31
B	43.06	45.07	56.88	76.94
C	45.83	42.85	40.42	62.08
D	58.06	63.19	57.71	59.72
E	91.39	95.35	89.93	96.67

Table 15 Computational time of different feature extraction methods for the TEP dataset

Feature Extraction Method	Computational time (s)
PCA	1.359375
ICA	1.578125
KPCA	163.609375
1D CNN	60.437500

achieve higher accuracy perfromances for bearing fault detection.

4.2.5 Comparison with traditional multi-classification methods

In this section, we further compare our method with three different multi-classification methods, namely, naïve Bayes (NB), random forest (RF) and support vector machine (SVM). In doing so, we conducted experiments using the five groups (A-E) of data described in Table 12. Taking group A as an example for explanation,the experimental settings are as follows.

1) Fault 2–5, 4–13, 15 are treated as known (seen) types, and faults 1, 6 and 14 are treated to be unknown (unseen). We denote the former by G1, and the latter by G2.
2) For the three traditional methods, samples from both G1 and G2 were were used to train the fault classification models. Specifically, a total of 480 × 12 = 5760 samples from G1 (480 samples for each fault type) were used for model training. In additional to the 5750 samples, two sets of samples from G2, one set conssiting of 10 × 3 = 30 (10 samples for each fault type) and another consisting of 50 × 3 = 150 (50 samples for each fault type), were also added to the training data for classification model training.
3) For the proposed method, only the 5760 samples from G1were used for SSB-ZSL-1DCNN model training; no sample from G2 was used.
4) A total of 480 × 3 = 1440 samples from G2 (480 samples for each fault type), which are the same as that used in the previous experiments, were used for model prfromance test.
5) The experiments for the three methods, NB, RF, and SVM, were implemneted by utilizing the scikit-learn package [47].

The results of the traditional multi-classification methods under few-shot learning setting are shown in Table 16.

From the results, all the three traditional multi-classification methods showed poor performance, even though a number of unseen fault samples were included in the training dataset. Further calculations show that the overall accuracies are distributed between 20% and 40%, the sample size of unseen faults being increased from 10 to 50. None of the three traditional methods produced a comarable result to that by our method. The highest accuracy of the three methods is 46.64%, which is achived by RF for Group E with 10 unseen fault samples being included in the training dataset. Our method achieves 96.67% Group E without using any samples in unseen faultsin five groups.

It shoulld be stressed that a diect comparison of our methods with these three traditional multi-classification methods is unfair and puts our method in a disadvantegous position, this is because zero-shot learning setting is totally different from few-shot learning setting and far more different

Table 16 Performance comparison of different multi-class classification methods for the TEP dataset (%)

Group.	Number of samples for each test fault type included in the training dataset						
	10			50			0
	NB	RF	SVM	NB	RF	SVM	SSB-ZSL-1DCNN
A	21.29	20.82	24.61	21.63	22.36	25.92	84.31
B	30.81	42.36	35.86	30.56	45.15	35.92	76.94
C	29.46	41.68	3354	29.68	44.63	34.14	62.08
D	26.58	29.64	30.26	30.17	33.96	33.43	59.72
E	26.79	46.64	33.43	27.79	46.24	33.39	96.67

than the settings for traditional multi-class classfication methods. It should be noted that the fault descriptions proposed in this study may not be applicable in the multi-classification methods. In conclusion, it is important to introduce fault attributes as the side information for fault detection under zero-shot learning setting.

4.2.6 Comparison with other ZSL methods

The experimental implementation process in this section is similar to that in Section 4.1.4, which considers four classic semantic based ZSL methods (i.e., SJE, DEVISE, ESZSL, SAE). We replace the visual attributes used in these four methods with the fault attributes proposed in this study. The experimental results are shown in Table 17, from which it can be seen that for most groups, our method outperforms the other four ZSL methods, except for group D. SAE is a deep learning-based method which uses auto-encoders with several hidden layers. It has a better non-linear representation ability than the other three shallow network models. For group D, the accuracy of SSB-ZSL-1DCNN is slightly lower than that of SAE (by 5.84%), but for other groups, our method performs much better than SAE: 38.61% higher for group A, 2.64% higher for group B, 28.54% higher for group C and 10.16% higher for group E. Overall, SSB-ZSL-1DCNN is more suitable for 1D time-series data thanks to the introduction of the fault attribute matrix. This, in turn, shows the importance and usability of feature extraction and fault descriptions for solving zero-shot fault detection problem.

5 Conclusion

In the field of fault detection, we often encounter the following situation: samples of certain types of faults are not available or extemely difficult to obtain for various reasons. Bearing this in mind, a new semantic space based zero-shot learning model with 1D CNN (SSB-ZSL-1DCNN) is proposed in this work for fault detection. The proposed method has the following feature: a SSB-ZSL-1DCNN model can detect new (unseen) faults even though the dataset used for training the model does not include any samples of the unseen faults. This is important and useful for solving fualt detection tasks where new fault types that have neve been seen before.

Table 17 Performance comparison of different zero-shot learning method for fault detection of the TE dataset (%)

Group.	SJE	DEVISE	ESZSL	SAE	SSB-ZSL-1DCNN
A	54.44	50.63	70.21	45.70	84.31
B	74.17	74.17	33.68	74.30	76.94
C	38.54	32.85	38.75	33.54	62.08
D	61.11	51.53	31.46	65.56	59.72
E	94.78	95.62	91.81	86.53	96.67

The applicablity and effectiveness of the prososed SSB-ZSL-1DCNN has been demonstrated using two well-known benchmarks: the bearing dataset CWRU for rotary machines and the TEP dataset for an entire chemical process control system. For the first case, the experiments focus on training the model using only samples of small size faults and detecting the large-size faults using the trained model. For the second case, the focus is on solving many-fault detection and many-class classfication problems, which are more comprehensive and challenging tasks. For both case studies, the proposed method shows excellent and impressive performances, which are far better than the compared methods.

In the future, we will improve the overall performance of the proposed SSB-ZSL-1DCNN model from the following two aspects. Firstly, the utility of the source of side information could be extended to unsupervised fault label embeddings rather than defining fault attributes manually; this can help increase the efficiency of the method. Secondly, the application of the proposed method can be extended to generalized zero-shot learning (GZSL), which emphasizes a more practical scenario where there is a need to effectviely detect both seen (known) faults and unseen (unknown) faults by training models using only samples of seen faults.

Acknowledgements Siyu Zhang gratefully acknowledges a scholarship from the Chines Scholarship Council (CSC). The authors acknowledge that this work was supported in part by NERC (Ref. NE/V002511/1), EPSRC (Ref. EP/H00453X/1) and EPSRC (Ref. EP/I011056/1).

References

1. Zhang T et al (2022) Intelligent fault diagnosis of machines with small & imbalanced data: a state-of-the-art review and possible extensions. ISA Trans 119:152–171. https://doi.org/10.1016/j.isatra.2021.02.042
2. Gertler JJ (2017) Fault detection and diagnosis in engineering systems. CRC press. https://doi.org/10.1201/9780203756126
3. Feng L, Zhao C (2021) Fault description based attribute transfer for zero-sample industrial fault diagnosis. IEEE

Trans Industr Inf 17(3):1852–1862. https://doi.org/10.1109/tii.2020.2988208

4. Sun C, Ma M, Zhao Z, Tian S, Yan R, Chen X (2018) Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. IEEE Trans Industr Inf 15(4): 2416–2425
5. Romera-Paredes B, Torr P (2015) An embarrassingly simple approach to zero-shot learning. In: International conference on machine learning. PMLR, pp 2152–2161
6. Akata Z, Perronnin F, Harchaoui Z, Schmid C (2013) Label-embedding for attribute-based classification. Presented at the 2013 IEEE conference on computer vision and pattern recognition
7. Pan J, Zi Y, Chen J, Zhou Z, Wang B (2017) LiftingNet: a novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification. IEEE Trans Industr Electron 65(6):4973–4982
8. Abdi H, Williams LJ (2010) Principal component analysis. Wiley Interdiscip Rev Comput Stat 2(4):433–459
9. Lee T-W (1998) Independent component analysis. In: Independent component analysis. Springer, pp 27–66
10. Lee J-M, Yoo C, Choi SW, Vanrolleghem PA, Lee I-B (2004) Nonlinear process monitoring using kernel principal component analysis. Chem Eng Sci 59(1):223–234
11. Gu X, Zhao Y, Yang G, Li L (2022) An imbalance modified convolutional neural network with incremental learning for chemical fault diagnosis. IEEE Trans Industr Inf 18(6):3630–3639. https://doi.org/10.1109/tii.2021.3112988
12. Hsueh Y-M, Ittangihal VR, Wu W-B, Chang H-C, Kuo C-C (2019) Fault diagnosis system for induction motors by CNN using empirical wavelet transform. Symmetry 11(10):1212
13. Paul D, Mohanty SK (2019) Fault classification in transmission lines using wavelet and CNN. In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT). IEEE, pp 1–6
14. Shen S et al (2021) A physics-informed deep learning approach for bearing fault detection. Eng Appl Artif Intell 103. https://doi.org/10.1016/j.engappai.2021.104295
15. Xu G, Liu M, Jiang Z, Söffker D, Shen W (2019) Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. Sensors 19(5):1088
16. Patel VM, Gopalan R, Li R, Chellappa R (2015) Visual domain adaptation: a survey of recent advances. IEEE Signal Process Mag 32(3):53–69
17. Zhang W, Li C, Peng G, Chen Y, Zhang Z (2018) A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. Mech Syst Signal Process 100:439–453
18. Wen L, Gao L, Li X (2017) A new deep transfer learning based on sparse auto-encoder for fault diagnosis. IEEE Trans Syst Man Cybern Syst 49(1):136–144
19. Liu Z-H, Jiang L-B, Wei H-L, Chen L, Li X-H (2021) Optimal transport-based deep domain adaptation approach for fault diagnosis of rotating machine. IEEE Trans Instrum Meas 70:1–12
20. Liu Z-H, Lu B-L, Wei H-L, Chen L, Li X-H, Rätsch M (2019) Deep adversarial domain adaptation model for bearing fault diagnosis. IEEE Trans Syst Man Cybernetics: Syst 51(7):4217–4226
21. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
22. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 951–958
23. Wang W, Zheng VW, Yu H, Miao C (2019) A survey of zero-shot learning. ACM Trans Intell Syst Technol 10(2):1–37. https://doi.org/10.1145/3293318
24. Song J, Shen C, Yang Y, Liu Y, Song M (2018) Transductive unbiased embedding for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1024–1033
25. Pang Y, Wang H, Yu Y, Ji Z (2019) A decadal survey of zero-shot image classification. Sci Sin Inf 49(10):1299–1320. https://doi.org/10.1360/n112018-00312
26. Lampert CH, Nickisch H, Harmeling S (2013) Attribute-based classification for zero-shot visual object categorization. IEEE Trans Pattern Anal Mach Intell 36(3):453–465
27. Xian Y, Schiele B, Akata Z (2017) Zero-shot learning-the good, the bad and the ugly. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4582–4591
28. Xian Y, Lampert CH, Schiele B, Akata Z (2019) Zero-shot Learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE Trans Pattern Anal Mach Intell 41(9):2251–2265. https://doi.org/10.1109/TPAMI.2018.2857768
29. Frome A et al (2013) Devise: a deep visual-semantic embedding model. Adv Neural Information Process Syst 26:2121–2129
30. Akata Z, Reed S, Walter D, Lee H, Schiele B (2015) Evaluation of output embeddings for fine-grained image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2927–2936
31. Kodirov E, Xiang T, Gong S (2017) Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3174–3183
32. Xian Y, Akata Z, Sharma G, Nguyen Q, Hein M, Schiele B (2016) Latent embeddings for zero-shot classification. Presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
33. Changpinyo S, Chao W-L, Gong B, Sha F (2016) Synthesized classifiers for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5327–5336
34. Hayashi T, Fujita H (2021) Cluster-based zero-shot learning for multivariate data. J Ambient Intell Humaniz Comput 12(2):1897–1911
35. Hayashi T, Fujita H, Hernandez-Matamoros A (2021) Less complexity one-class classification approach using construction error of convolutional image transformation network. Inf Sci 560:217–234
36. Lv H, Chen J, Pan T, Zhou Z (2020) Hybrid attribute conditional adversarial denoising autoencoder for zero-shot classification of mechanical intelligent fault diagnosis. Appl Soft Comput 95: 106577
37. Gao Y, Gao L, Li X, Zheng Y (2020) A zero-shot learning method for fault diagnosis under unknown working loads. J Intell Manuf 31(4):899–909
38. Xing S, Lei Y, Wang S, Lu N, Li N (2022) A label description space embedded model for zero-shot intelligent diagnosis of mechanical compound faults. Mech Syst Signal Process 162:108036. https://doi.org/10.1016/j.ymssp.2021.108036
39. Xu J, Zhou L, Zhao W, Fan Y, Ding X, Yuan X (2022) Zero-shot learning for compound fault diagnosis of bearings. Expert Syst Appl 190. https://doi.org/10.1016/j.eswa.2021.116197
40. Downs JJ, Vogel EF (1993) A plant-wide industrial process control problem. Comput Chem Eng 17(3):245–255
41. Barshan E, Ghodsi A, Azimifar Z, Jahromi MZ (2011) Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. Pattern Recogn 44(7): 1357–1371
42. Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ (2021) 1D convolutional neural networks and applications: a survey. Mech Syst Signal Process 151:107398
43. Weston J, Bengio S, Usunier N (2010) Large scale image annotation: learning to rank with joint word-image embeddings. Mach Learn 81(1):21–35
44. Smith WA, Randall RB (2015) Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark

study. Mech Syst Signal Process 64–65:100–131. https://doi.org/10.1016/j.ymssp.2015.04.021
45. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
46. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
47. Pedregosa F et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Siyu Zhang received the B.Eng. degree in automation from Northeastern University, Shenyang, China, in 2020, and MRes degree in Control & Systems Eng from The University of Sheffield. Currently, she is working toward the M.Sc. degree in control science and engineering from Northeastern University.

Her current research interests include machine learning, zero-shot learning, fault detection, and feature extraction.

Jinliang Ding received the Bachelor, Master, and Ph.D. degrees in Control Theory and Control Engineering from Northeastern University, Shenyang, China, respectively.

He is a Professor of the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University. He has authored or co-authored over 200 refereed journal papers and refereed papers at international conferences. He has also invented or co-invented over 50 patents. His research interests include modeling, plant-wide control and optimization for the complex industrial systems, machine learning, industrial artificial intelligence, and computational intelligence and application.

Dr. Ding was a recipient of the Young Scholars Science and Technology Award of China in 2016, the National Science Fund for Distinguished Young Scholars in 2015, the National Technological Invention Award in 2013, and three First-Prize of Science and Technology Awards of the Ministry of Education in 2006, 2012, and 2018, respectively. One of his articles published on Control Engineering Practice was selected for the Best Paper Award of 2011–2013.