Mechanical Systems and Signal Processing 99 (2018) 459–477

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features

H.O.A. Ahmed [a], M.L.D. Wong [b], A.K. Nandi [a,c,*]

[a] Department of Electronic and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, United Kingdom
[b] Heriot-Watt University Malaysia, Precinct 5, 62200 Putrajaya, Malaysia
[c] The Key Laboratory of Embedded Systems and Service Computing, College of Electronic and Information Engineering, Tongji University, Shanghai, China

ARTICLE INFO

Article history:
Received 30 July 2016
Received in revised form 26 April 2017
Accepted 20 June 2017
Available online 4 July 2017

Keywords:
Compressed sensing
Sparse over-complete representations
Deep neural network
Sparse autoencoder
Bearing fault classification
Machine condition monitoring

ABSTRACT

Condition classification of rolling element bearings in rotating machines is important to prevent the breakdown of industrial machinery. A considerable amount of literature has been published on bearing faults classification. These studies aim to determine automatically the current status of a roller element bearing. Of these studies, methods based on compressed sensing (CS) have received some attention recently due to their ability to allow one to sample below the Nyquist sampling rate. This technology has many possible uses in machine condition monitoring and has been investigated as a possible approach for fault detection and classification in the compressed domain, i.e., without reconstructing the original signal. However, previous CS based methods have been found to be too weak for highly compressed data. The present paper explores computationally, for the first time, the effects of sparse autoencoder based over-complete sparse representations on the classification performance of highly compressed measurements of bearing vibration signals. For this study, the CS method was used to produce highly compressed measurements of the original bearing dataset. Then, an effective deep neural network (DNN) with unsupervised feature learning algorithm based on sparse autoencoder is used for learning over-complete sparse representations of these compressed datasets. Finally, the fault classification is achieved using two stages, namely, pre-training classification based on stacked autoencoder and softmax regression layer form the deep net stage (the first stage), and re-training classification based on backpropagation (BP) algorithm forms the fine-tuning stage (the second stage). The experimental results show that the proposed method is able to achieve high levels of accuracy even with extremely compressed measurements compared with the existing techniques.

1. Introduction

Rolling element bearings are among the most fundamental elements in rotating machinery and their failures are accountable for more substantial failures in the machine. Thus, roller bearings require an effective condition monitoring and machinery maintenance program to avoid machine breakdowns. In fact, machines' availability and functions can be monitored by

* Corresponding author at: Department of Electronic and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, United Kingdom.
E-mail address: asoke.nandi@brunel.ac.uk (A.K. Nandi).

http://dx.doi.org/10.1016/j.ymssp.2017.06.027
0888-3270/© 2017 The Authors. Published by Elsevier Ltd.

accurate, rapid and automatic fault detection techniques. Therefore, rolling element bearing condition monitoring (CM) [1] has attracted considerable attention from researchers in the past decades. In the course of their operation, rotating machines produce signals in different forms, e.g. noise, vibration, temperature, lubricating oil condition, etc. [2]. Various characteristic features can be observed from vibration signals that make it the best choice for machine condition monitoring. Vibration signal analysis can be performed in three main groups – time domain, frequency domain, and time-frequency domain analysis [3–6]. Time domain techniques extract features from the raw vibration signals using some statistical parameters, these include, peak-to-peak value, root mean square, crest Factor, skewness, kurtosis, Impulse factor, etc. [7]. The frequency domain analysis techniques have the ability to divulge some information based on the frequency characteristics that are not easily observed in time-domain. In practice, the time-domain signal is transformed into frequency-domain by using Fast Fourier Transform (FFT). The time-frequency domain has been used for non-stationary waveform signals which are very common when machinery fault occurs. Thus far, several time–frequency analysis techniques have been developed and applied to machinery fault diagnosis, e.g., Wavelet Transform (WT), Short Time Fourier Transform (STFT) adaptive parametric time-frequency analysis based on atomic decomposition, and non-parametric time-frequency analysis, including, Hilbert-Huang Transform (HHT), local mean decomposition, energy separation and empirical mode decomposition [8–15]. Spectral Kurtosis (SK) has been used effectively in the vibration-based condition monitoring of rotating machines. In this method, signal is first decomposed into the time-frequency domain where the kurtosis values are defined for each frequency group. In addition, Kurtogram concept has been proposed to increase the signal-to- noise ratio [16], and have been effectively used for both vibration and acoustic emission [17]. An alternative framework to strictly stationary vibration signal processing methods, cyclostationary analysis has been used for analysing vibration signals [18,19].

All the above techniques use data that have been recorded satisfying the Shannon/Nyquist sampling theorem, in which the sampling rate must be at least double the maximum frequency present in the signal. One aspect of much of the literature on using Nyquist sampling rate is that it may result in measuring a large amount of data. It is clear that acquiring a large amount of data requires large storage and time for signal processing and this also may limit the number of machines that can be monitored remotely across wireless sensor networks (WSNs) due to bandwidth and power constraints.

Compressive sensing (CS) also called compressed sensing or compressed sampling [20,21] is a new technique that supports sampling below the Nyquist rate. CS is being considered in a large diversity of applications including medical imaging, seismic imaging, and radio detection and ranging, communications and networks [22–27]. The basic idea of CS is that original signals can be reconstructed from fewer measurements far below than the Shannon sampling rate using sparse representation and a well-designed measurement matrix. The literature on compressive sensing shows a variety of approaches to reconstruct signals from few measurements [28–31]. In an analysis of effects of compressive sensing on the classification of bearing faults after reconstructing the original signal, Wong et al. [32] found slight performance degradation with a large reduction of bandwidth constraint caused by using CS. Similarly, Li et al. in [33] shown the possibility to detect the fault of train's rolling bearing from the reconstructed signal based on compressive sensing. However, signal reconstruction techniques may not be practical in all applications and make no attempt to address the question of whether or not it is possible to learn in the compressed domain rather than recovering original signals. For instance, bearing vibration signal is always acquired for faults detection and estimation, and as long as it is possible to detect faulty signals in the measurement domain, then it is not necessary to recover the original signal to identify faults. A significant analysis and discussion on the subject of how to solve a range of signal detection and estimation problems given compressed measurements without reconstructing the original signal was presented by Davenport et al. in [34].

Over the past years, most research in compressive sensing based methods has emphasized the use of compressed measurements, sparse representations and incomplete signal reconstruction for the bearing fault diagnosis. For example, Tang et al. [35] developed a sparse classification strategy based on compressive sensing by extracting and classifying fault features through sparse representation combined with random dimensionality reduction. Zhang et al. [36] suggest a bearing fault diagnosis method based on the low-dimensional compressed vibration signal by training several over-complete dictionaries that can be effective in signal sparse decomposition for each vibration signal state. Another learning dictionary basis for extracting impulse components is described by Chen et al. in [37]. Tang et al. [38] proposed an interesting approach in which authors attempted to observe the characteristic harmonics from sparse measurements through a compressive matching pursuit strategy during the process of incomplete reconstruction. The outcomes of these studies corroborate the efficiency of compressive sensing in machinery fault diagnosis.

Even though the efficiency of CS in machine fault diagnosis has been validated in these studies, most of the fault classification improvements in the literature were achieved by increasing the sampling rate, otherwise highly compressed measurements attained poor classification accuracy. The aim of this work is to improve the efficiency of using highly compressed measurements for signal classification purpose. With this goal, this work explores the possibility of learning sparse overcomplete representations from highly compressed measurements in an unsupervised manner based on a deep learning approach.

Various deep learning architectures, e.g., Convolution Deep Neural Networks (CNNs), Deep Belief Networks (DBNs), Recurrent Neural Networks (RNNs), and stacked Autoencoders (AEs), have been used for reducing dimensionality or extracting features from signals. Unlike the standard Neural Network (NN), the architecture of CNN is usually composed of a convolutional layer and a sub-sampling layer, also called a pooling layer. CNN learns abstract features by alternating and stacking convolutional layers, and pooling operation. The convolution layers convolve multiple local filters with raw input data and generate invariant local features and the pooling layers extract most significant features. DBNs are generative neural networks that stack multiple Restricted Boltzmann Machines (RBM) that can be trained in a greedy layer-wise unsupervised way, then it can be further fine-tuned with respect to labels of training data by adding a softmax layer in the top layer. RNN builds connections between units from a direct cycle and map form the entire history of previous inputs to target vectors in principal and allows a memory of previous inputs to be kept in the network state. As is the case with DBNs, RNNs can be trained via backpropagation through time for supervised tasks with sequential input data and target outputs. A good overview of these deep learning architectures can be found in [39].

Previous research has shown that sparse representations of signals are able to signify the diagnosis features for machinery fault [40–44]. The advantages of sparse over-complete representations i.e. the number of obtained features is greater than the number of input samples have been studied by Lewicki et al. [45], who concluded that over-complete bases can produce a better approximation of the underlying statistical distribution of the data. Olshausen et al. [46] and Doi et al. [47] identify several advantages of over-complete basis set, for example, their robustness to noise and their ability to improve classification performance. Sparse feature learning methods normally contain two stages: (1) produce a dictionary W that represents the data $\{x_i\}_{i=1}^{N}$ sparsely using a learning algorithm, e.g. training artificial neural network (ANN) with sparsity penalties; and (2) obtain a feature vector from a new input vector using encoding algorithm.

Various recent studies investigating sparse feature representation have been carried out. These include: Sparse Autoencoder (SAE) [48] Sparse Coding [43] and RBMs [49]. SAE approach has a number of attractive features: (1) simple to train, (2) the encoding stage is very fast, and (3) the ability to learn features when the number of hidden units is greater than the number of input samples. Therefore it was decided that SAE is an appropriate method to adopt for our investigation. In an analysis of Autoencoder (AE), Bengio et al., [50] found that AE can be used as a building block of a Deep Neural Network (DNN), using greedy layer-wise pre-training.

Several studies have been undertaken to use deep neural network using an autoencoder algorithm for the purpose of machinery fault diagnosis. For example, Tao et al. [51] suggested a deep neural network algorithm framework for bearing fault diagnosis based on stacked Autoencoder and softmax regression. Jia et al. in [52] shown the effectiveness of a proposed DNN-based intelligent method in the classification of different datasets from rolling element bearings and planetary gearboxes with massive samples using Autoencoder as a learning algorithm. In a recent paper by Sun et al. [53], a sparse Autoencoder-based deep neural network approach with the help of the denoising coding and dropout method using one hidden layer was proposed for induction motor fault classification with 600 data samples and 2000 features from each induction motor working condition. The results of these investigations validate the effectiveness of DNN based on autoencoder learning algorithm in machinery fault classification. However, their focus was mainly on using autoencoder as a dimensionality reduction technique, i.e., the number of hidden nodes in each hidden layer is less than the number of input samples for the purpose of fault diagnosis with large amount of input data.

This paper proposes a novel intelligent classification method for bearing faults from highly compressed measurements using sparse-over-complete features and training DNN through SAE. In this method we impose some flexible constraints for regularization on the hidden units of the sparse-Autoencoder. These include sparsity constraint that can be controlled by different parameters such as, sparsity parameter, weight decay parameter and the weight of the sparsity penalty parameter. For the intent of learning sparse over-complete representations of our highly compressed measurements the number of hidden units in each hidden layer is set to be greater than the number of input samples, and we used the encoder part of our unsupervised learning algorithm (i.e. The SAE). One important aspect in this proposed method is to pre-train the DNNs in an unsupervised manner using previously described SAE and then to fine-tune it with back-propagation (BP) algorithm for classification. The difficulty of multilayer training can be overcome with an appropriate set of features. One of the advantages of pre-training and fine-tuning in this approach is the power to mine flexibly fault features from highly compressed signals. Thus, the proposed approach is expected to achieve better classification accuracy compared with methods based on under-complete feature representations. Consequently, the efficiency of compressive sensing in machine faults classification is expected to be improved.

The remainder of this paper is organized as follows. Section 2 briefly describes the theoretical background of CS. Section 3 introduces the process of DNN for bearing faults classification, with SAE algorithm and softmax regression. Section 4 is devoted to descriptions of the proposed method. Section 5 is dedicated to a description of the performed experiments and datasets used in these experiments using different sampling rates and the corresponding experimental results. In addition, the proposed method is compared with several published methods using the same datasets. Finally, Section 6 draws some conclusions from this study.

2. Compressive sensing

This section gives a brief description of CS [20,21]. Quite recently, considerable attention has been paid to compressive sensing for its ability to allow one to sample far below the Nyquist sampling rate and yet be able to reconstruct the original signal when needed. The basic thought is that many real-world signals have sparse features in some domain, e.g., wavelet transform, can be reconstructed from fewer measurements under certain conditions. Suppose we have n data points, $x_i \in R^N$, we call these data points the original signal. To produce a set of sparse components of these data points we need to use sparsifying transform ψ and this can be computed by the following equation:

$$x = \psi s \tag{1}$$

where s is $N * 1$ column vector with k nonzero coefficients and represent the sparse elements. According to compressive sampling theory, the signal x can be recovered from its compressed measurements y when the measurement matrix ϕ is incoherent with the dictionary (sparsifying transform) ψ such that

$$y = \phi\psi s \tag{2}$$

Founded along the idea of compressive sensing, when Φ and Ψ are incoherent, the original signal can be recovered from m = O (K log (N)) Gaussian measurements or $K \le C.m/\log(N/m)$ Bernoulli measurements [54]. Random matrix with i.i.d. Gaussian entries or Bernoulli (±1) matrix are both satisfy Restricted Isometry Property (RIP) [55].

Definition 1.1. The measurement matrix ϕ satisfies the Restricted Isometry Property (RIP) if there exists a parameter $\delta \in (0,1)$ such that

$$(1-\delta)\|s\|_2^2 \leqslant \phi\|x\|_2^2 \leqslant (1+\delta)\|s\|_2^2 \tag{3}$$

holds for all sparse vectors s.

Then we can take $\phi \epsilon R^{mxN}$ to be one of these random matrices, the size of this matrix $(m * N)$ depends on the compressive sampling rate (α), which is significantly lower than the Nyquist rate $(m \ll N)$. In this case, Eq. (2) is underdetermined. To estimate the coefficients vector s is to solve the optimization problem by using $\ell 1$-norm; thus, our estimation $\hat{s}$ of the original s is given by the equation

$$\hat{s} = \min_{s \in R^N} \frac{1}{2}\|\phi\psi s - y\|_{l2}^2 + \gamma\|s\|_{l1} \tag{4}$$

where $\|\phi\psi s - \mathrm{y}\|_{l2}^2 \leqslant \varepsilon$ for a chosen $\varepsilon > o$, and a particular regularization parameter $\gamma > 0$ that controls the relative importance applied to the sparseness $\ell 1$ and the $\ell 2$ error. Based on CS we can reconstruct the original signal x from the sparsifying transform (ψ) and $\hat{s}$ such that

$$\hat{x} = \psi\hat{s} \tag{5}$$

The set of compressed data y provided by the compressive sensing framework in Eq. (2) has enough information to reconstruct the original signal. In our case, this compressed measurement can be applied directly for faults classification.

3. Deep neural network

Although, supervised learning based artificial neural network (ANN) with many hidden layers are found to be difficult in practice, DNNs have been well developed as a research topic and have been made practically feasible with the assistance of unsupervised learning. Moreover, DNNs have attracted extensive attention by outperforming other machine learning methods. Each layer of DNN performs a non-linear transformation of the input samples in the preceding layer to the following one. A good overview of DNNs can be found in [56]. Different from ANN, DNNs can be trained in a supervised or unsupervised manner [50,57] and they are also appropriate in the general area of Reinforcement Learning (RL) [58,59]. The basic idea of training DNN is that we first train the network layer by layer using unsupervised learning algorithm, e.g. autoencoder; this process is called DNN pre-training. In this process, the output from each layer will be the input to the succeeding layer. Then the DNN is retrained in a supervised way with backpropagation algorithm for classification.

3.1. Sparse autoencoder

An autoencoder neural network provides a means of an unsupervised learning algorithm that sets the target values, i.e., the outputs to be equal to the inputs and applies backpropagation [48]. As shown in Fig. 1, like many unsupervised feature learning methods the design of an autoencoder relies on an encoder - decoder architecture, where the encoder produces a feature vector from the input samples and the decoder recovers the input from this feature vector. The encoder part is a feature extraction function, f_θ, that compute a feature vector $h\ (x_i)$ from an input x_i, we define

$$H(x_i) = f_\theta(x_i) \tag{6}$$

where, $h(x_i)$ is the feature representation. The decoder part is a recovery function, g_θ, that reconstructs the input space $\widetilde{x}_i$ from the feature space $h(x_i)$ such that

$$\widetilde{x}_i = g_\theta(h(x_i)) \tag{7}$$

The autoencoder is attempting to learn an approximation such that x_i is similar to $\widetilde{x}_i$, i.e., is trying to attain the lowest possible reconstruction error $E\ (x_i, \widetilde{x}_i)$ that measure the discrepancy between x_i and $\widetilde{x}_i$. Hence the following equation is obtained

$$E(x_i, \widetilde{x}_i) = \|x_i - \widetilde{x}_i\|^2 \tag{8}$$

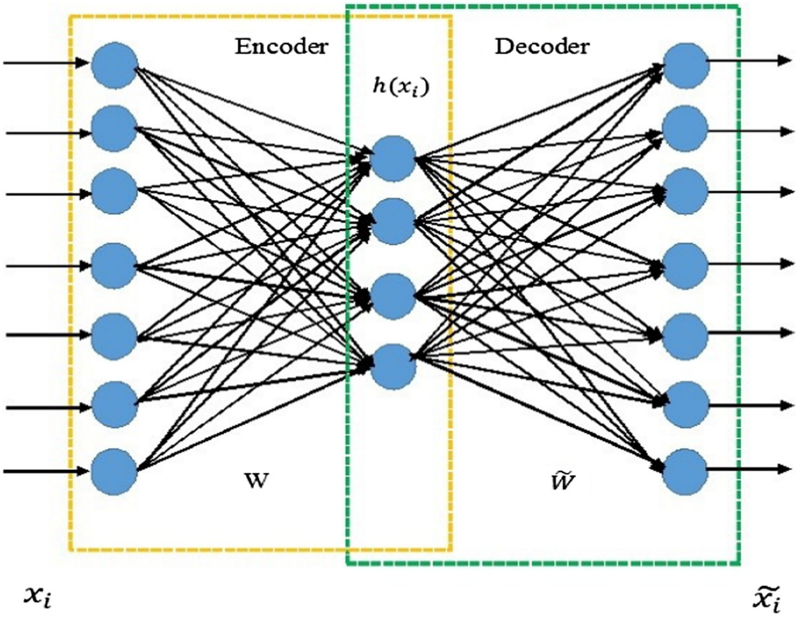

Fig. 1. Autoencoder architecture.

In fact, autoencoders were mainly developed as a multi-layer perceptron (MLP) and the most common used forms for the encoder and decoder are affine transformations that keep collinearity followed by a nonlinearity:

$$f_\theta(x_i) = s_f(b + W) \tag{9}$$

$$g_\theta(h(x_i)) = s_g(c + \widetilde{W}) \tag{10}$$

where s_f and s_g are the encoder and decoder activation function, e.g. sigmoid and hyperbolic tangent, b and c are the encoder and decoder bias vectors, W and $\widetilde{W}$ are the encoder and decoder weight matrices.

The autoencoder is one of the more practical ways of reinforcement learning. For instance, by forcing some constraints on the autoencoder network such as limiting the number of hidden units and imposing some regularizers, Autoencoder may learn interesting feature structure about the data. Therefore, different constraints give different forms of autoencoders. Sparse Autoencoder (SAE) is an autoencoder that contains sparsity constraint on the hidden unit's activation that must typically be near 0. This may be accomplished by adding Kullback-Leibler (KL) divergence penalty term

$$\sum_{j=1}^{d} KL(\rho \| \hat{\rho}) \tag{11}$$

where

$$KL(\rho // \hat{\rho}) = \rho \log \frac{\rho}{\widehat{\rho_j}} + (1 - \rho) \log \frac{1 - \rho}{1 - \widehat{\rho_j}} \tag{12}$$

and ρ is a sparsity parameter, normally its value can be small and close to zero, e.g., $\rho = 0.2$, while $\hat{\rho}$ is the average threshold activation of hidden units and can be calculated by the following equation:

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} [a_j^2(x_i)] \tag{13}$$

where a_j^2 represents the activation of hidden unit j. By minimizing this penalty term $\hat{\rho}$ would be close to ρ, and the overall cost function (CF) can be calculated by the following equation

$$CF_{sparse}(W, b) = \frac{1}{2m} \sum_{i=1}^{m} \|\widetilde{x}_i - x_i\|^2 + \lambda \|W\|^2 + \beta \sum_{j=1}^{d} KL(\rho \| \hat{\rho}) \tag{14}$$

where m is the input size, d is the hidden layer size, λ represents weight decay parameter, and β is the weight of the sparsity penalty term.

3.2. Softmax regression

Softmax regression also called multinomial logistic regression [60], is a supervised regression model that generalizes the logistic regression where labels are binary, i.e., $c^{(i)} \in \{0, 1\}$ to multi-classification problems that have labels $c^{(i)} \in \{1, \ldots, K\}$ where K is the number of classes. Briefly, we present the simplified softmax regression algorithm as follows:

Let a training set $\{(x^{(1)}, c^{(1)}), \ldots, (x^{(m)}, c^{(m)})\}$ of m labelled examples and input features$x^{(i)} \in R^n$. In logistic regression with binary labels, $c^{(i)} \in \{0, 1\}$ our hypothesis can be written as follows:

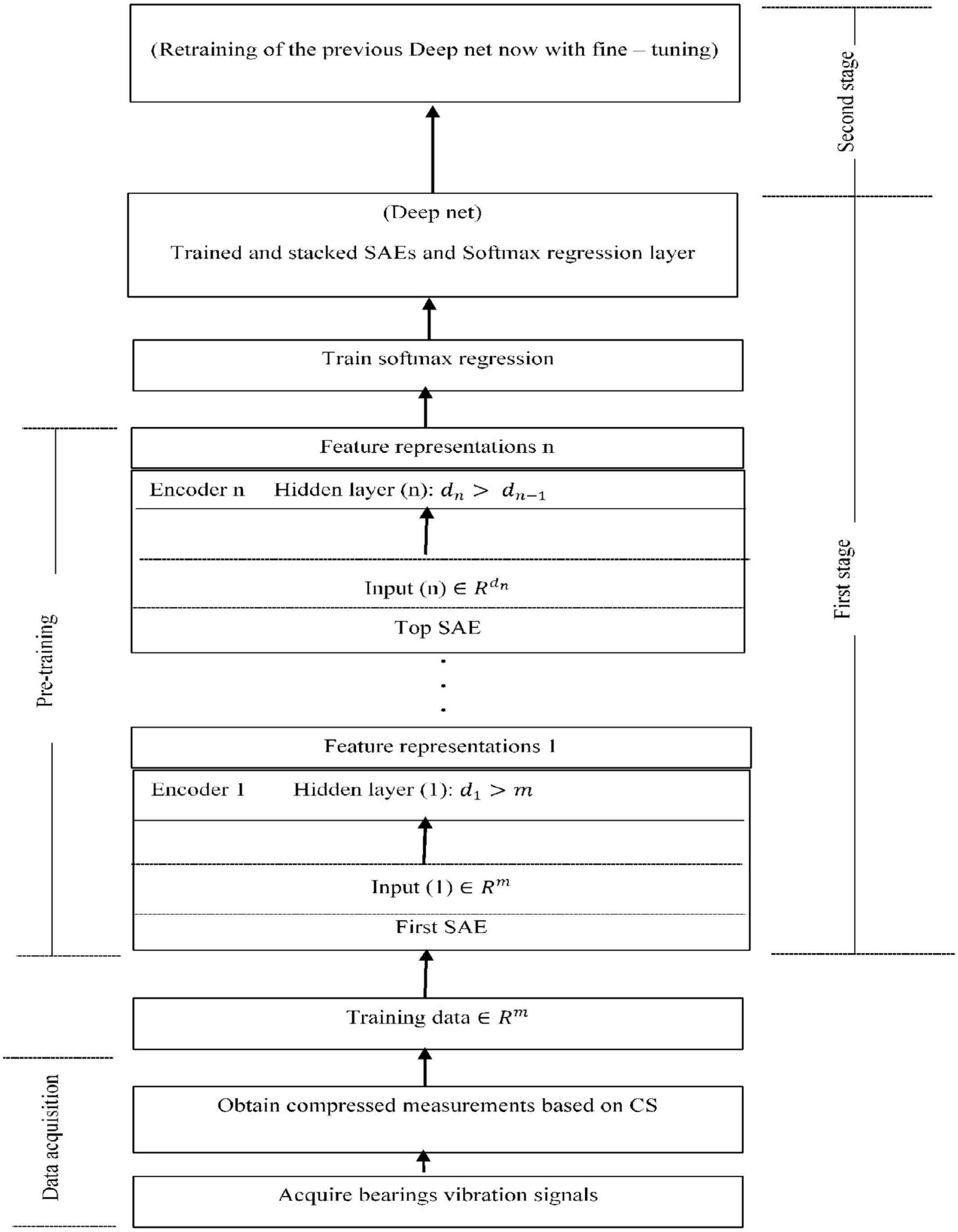
(Retraining of the previous Deep net now with fine – tuning)
Second stage
(Deep net)
Trained and stacked SAEs and Softmax regression layer
Train softmax regression
Feature representations n
Encoder n
Hidden layer (n): $d_n > d_{n-1}$
Input (n) $\in R^{d_n}$
Top SAE
First stage
Pre-training
Feature representations 1
Encoder 1
Hidden layer (1): $d_1 > m$
Input (1) $\in R^m$
First SAE
Training data $\in R^m$
Obtain compressed measurements based on CS
Acquire bearings vibration signals
Data acquisition

Fig. 2. Training of our proposed method.

$$h_0(x) = \frac{1}{1 + \exp(-\theta^T x)} \tag{15}$$

where θ are model parameters that are trained to minimize the cost function $J(\theta)$ defined by the following equation

$$J(\theta) = -\left[\sum_{i=1}^{m} c^{(i)} \log h_\theta(x^i) + (1 - c^{(i)}) \log(1 - h_\theta(x^i))\right] \tag{16}$$

In softmax regression with multi-labels $c^{(i)} \in \{1, \ldots, K\}$ our hypothesis will be to estimate the probability $P(c = k|x)$ for each value of k = 1 to K, such that

$$h_\theta(x) = \begin{bmatrix} P(c = 1|x;\theta) \\ P(c = 2|x;\theta) \\ \vdots \\ P(c = K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)T} x)} \begin{bmatrix} \exp(\theta^{(1)T} x) \\ \exp(\theta^{(2)T} x) \\ \vdots \\ \exp(\theta^{(K)T} x) \end{bmatrix} \tag{17}$$

where $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)} \in R^n$ are the parameters of softmax regression model.

4. Proposed algorithm

To predict the status of the rolling bearing from highly compressed measurements, a novel feature learning approach is proposed in this paper. The intention of this technique is to learn sparse-over-complete feature representations from these compressed measurements for the purpose of faults classifications. Our method applies a learning algorithm in multi-stages of non-linear feature transformation, each stage is a kind of feature transformation. One way to do this is by using DNN with multiple hidden layers and each layer is connected to the layers below it in a non-linear combination. In the pre-training stage, sparse-autoencoder is used to train the DNN, the encoder part of the sparse-autoencoder with sigmoid activation function was used to learn the over-complete feature representations. The sparse autoencoder particularly has many attractive aspects that make it an appropriate choice for our investigation, e.g., SAEs are simple to train and their encoding stage is very fast.

As expressed in Fig 2, the proposed method produces over-complete representations from the input compressed measurements (y) by setting the number of hidden units (d_i) to be greater than the number of input samples (m), i.e., $d_i > m$ in each hidden layer (i), where $d_{i+1} > d_i$ for $i = 1, 2, 3, \ldots, n$ and Input (n) represent the output of Encoder (n – 1), and d_n is the number of hidden layers in Encoder (n). As drawn in Section 3, DNN training includes two levels of training, namely, pre-training using unsupervised learning algorithm and re-training using backpropagation algorithm. In the pre-training stage, the unlabelled bearing compressed measurements (y) are first used to train DNN by setting the parameters in each hidden layer and compute the sparse-over-complete feature representations. In fact, in the DNN based on sparse autoencoder we are making use of the SAE algorithm applied multiple times through the network. Therefore, the output over-complete feature vector from the first encoder is the input of the second encoder.

Finally, the fault classification is achieved using two stages, namely, (1) pre-training classification based on stacked autoencoder and softmax regression layer which is the deep net stage (the first stage), and (2) re-training classification based on backpropagation (BP) algorithm and that is the fine-tuning stage (the second stage).

The pre-training process can be described in the following steps:

```
Given a DNN of $n$ hidden layers, the pre-training process using sparse autoencoder to learn over-complete features will
  be conducted on each layer.
   (1) Initialization, $m$ is the number of input samples, $n$ is the number of hidden layers,
     $d_0 = m, i = 1, 2, \ldots, n$
   (2) For $i = 1 to n$,
   (3) Set up the number of hidden units ($d_i$) to be greater than the input samples ($m$), i.e., $d_i > d_{i-1}$.
   (4) Set up sparsity parameter, weight decay parameter, weight of the sparsity penalty term and the maximum
  training epoch that achieve the lowest possible reconstruction error E ($x_i, \widetilde{x}_i$).
   (5) Use scale conjugate gradient training (SCG) algorithm for network training. By utilizing SCG, the learning rate is
  automatically adopted at each epoch and the average threshold activation of hidden units ($\hat{\rho}$) can be computed using
  Eq. (11).
   (6) Based on Eq. (12), compute the cost function.
   (7) Using the encoder part of the sparse autoencoder calculate the output over-complete feature vector $v_i$ and use it
  as the input of the following hidden layer.
   (8) $i = i + 1$.
   (9) Repeat steps 2, to 8 until $i = n$.
  (10) Use the over-complete feature vector of the last hidden layer $v_n$ to be the input of the softmax regression layer.
```

Fig 3 shows an illustration of the pre-training process using two hidden layers. With enforcement of sparsity constraints and by setting the number of units in each hidden layer to be greater than input samples, each autoencoder learns useful features of the compressed unlabelled training samples. The training process is performed by optimizing the cost function $CF_{sparse}(W,b)$ in Eq. (14). The optimization is performed using Scaled Conjugate Gradient (SCG) which is a member of Conjugate Gradient (CG) methods [61]. In the first learning stage, the encoder part of the first SAE with sigmoid functions of the range of unit activation function values [0,1] is used to learn features from compressed vibration signals of length m, where number of hidden units $d_1 > m$ and the extracted over-complete features (υ_1) are used as the input signals for the second learning stage. Then Encoder 2 of the second SAE with number of hidden units d_2 is used to extract over-complete features (υ_2) from (υ_1) Finally, softmax regression is trained using (υ_2) to classify bearings health conditions.

5. Experimental study

Automatic condition monitoring on rolling element bearings (Fig. 4) is essential to avoid machine breakdown. Given highly compressed measurements (recorded below the Nyquist sampling rate), our proposed method is designed to learn directly from these compressed measurements without recovering the original signal. In this section, two faults classification cases of rolling element bearings are used to validate the proposed method.

5.1. First case study

5.1.1. Data description

The vibration data used in this case were collected from experiments on a small test rig that simulates running roller bearings' environment. Six conditions of roller bearings status have been recorded and examined. These include, two normal conditions, namely, a brand new condition (NO) and a worn but undamaged condition (NW); and four fault conditions containing, inner race (IR) fault, an outer race (OR) fault, rolling element (RE) fault, and cage (CA) fault. Each condition has its corresponding unique characteristic as follows.

(1) The NO bearing is a brand new bearing and in perfect condition.
(2) The NW bearing is in service for some period of time but in good condition.
(3) The IR fault is created by first removing the cage, moving the elements to one side of the bearing, and then removing the inner race. A groove was cut in the raceway of the inner race using a small grinding stone, and the bearing was reassembled.
(4) The OR fault is created by removing the cage, pushing all the balls to one side, and then inserting a small grinding stone and cutting a small groove in the outer raceway.
(5) The RE fault is created by using an electrical etcher to mark the surface of one of the balls, simulating corrosion.
(6) The CA fault is created by removing the plastic cage from one of the bearings, cutting away a section of the cage so that two of the balls were free to move and not held at a regular spacing, as would normally be the case.

Data were recorded at 16 different speeds. Fig. 5 depicts some typical time series plots for the six different aforementioned conditions. Depending on the fault conditions, the defects modulate the vibration signals with their own patterns. The inner and outer race fault conditions have a fairly periodic signal; the rolling element fault may or may not be periodic, dependent upon several factors including the level of damage to the rolling element, the loading of the bearing, and also the track that the ball describes within the raceway itself. The cage fault generates a random distortion, which also depends on the degree of damage and the bearing loading.

Fig. 6 shows the test rig to collect the vibration data of bearings. The test rig consists of a DC motor driving the shaft through a flexible coupling, with the shaft supported by two Plummer bearing blocks. A series of damaged bearing were inserted in one of the Plummer blocks, and the resultant vibrations in the horizontal and vertical planes were measured using two accelerometers. The output from the accelerometers was fed back through a charge amplifier to a Loughborough Sound Images DSP32 ADC card (using a low-pass filter with a cut-off 18 kHz), and sampled at 48 kHz, giving a slight oversampling. The machine was run at a series of 16 different speeds ranging between 25 and 75 rev/s, and ten-time series were taken at each speed. This gave a total of 160 examples of each condition, and a total of 960 raw data files to work with. The description of the dataset is presented in Table 1.

5.1.2. Processing of data

We began by obtaining the compressed vibration signal from the big data of rolling elements bearing vibration signal. First, we used wavelet transform to decompose the signal into low and high-frequency levels to obtain the sparse components that are demanded by the compressive sensing framework. One of the choices is Haar wavelet basis that had been used as sparse representations for vibration signals in several research papers e.g., [62,63]. We used Haar wavelet basis with five decomposition levels as sparsifying transform. The wavelet coefficients of the vibration data are displayed in Fig. 7(a). After applying the penalized hard threshold [64], the wavelet coefficients are sparse in the Haar wavelet domain as shown in Fig. 7

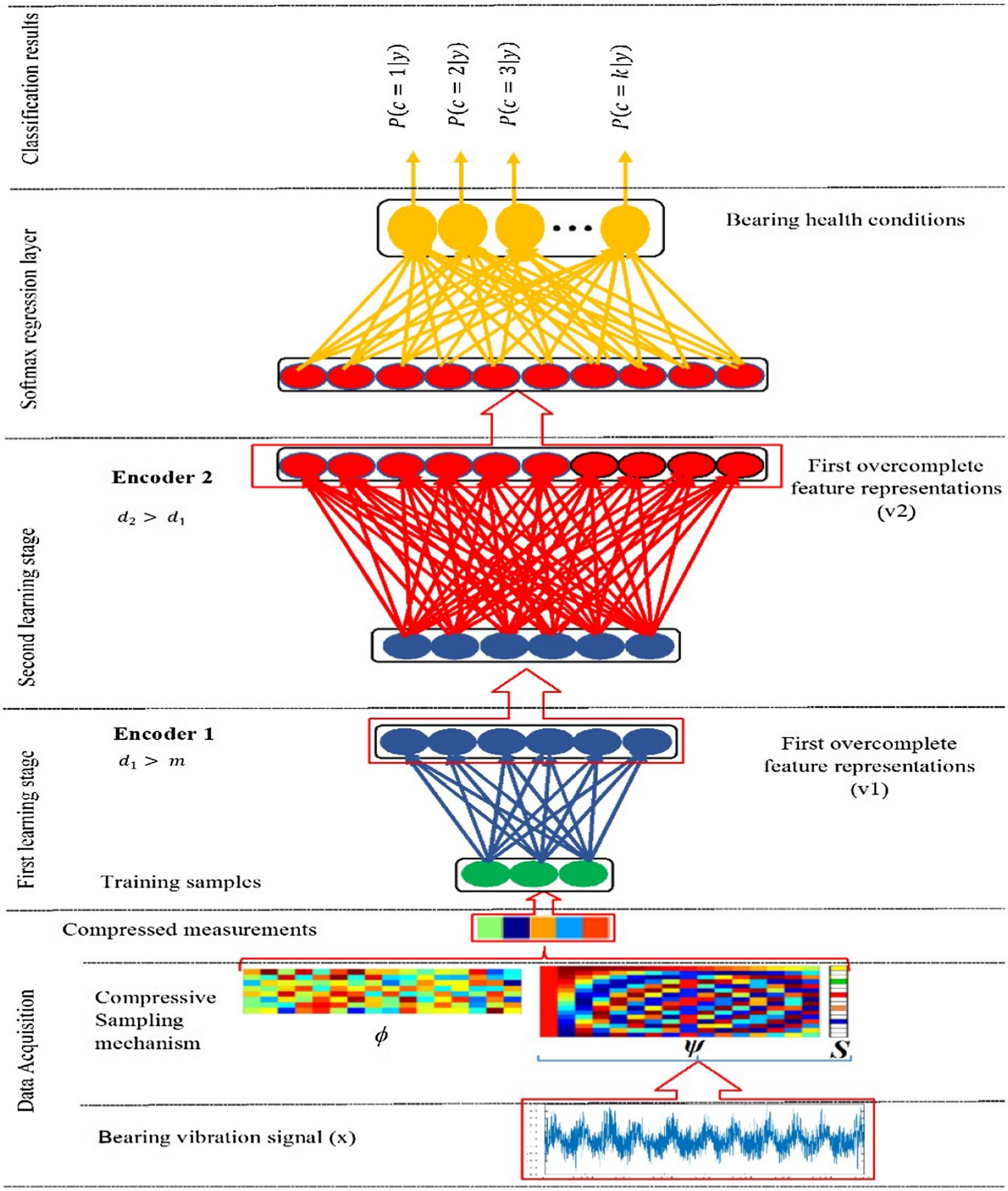

Fig. 3. Illustration of the proposed method using two hidden layers. Data flow from the bottom to the top.

(b) where only 216 are non-zeros (nnz) in NO wavelet coefficients that is 95.8% of the 5120 coefficients are zeros. Other conditions NW, IR, OR, RE and CA have 276, 209, 298, 199, and 299 non-zero elements respectively; and 94.6%, 95.9, 94.2%, 96.1 and 94.2% of 5120 coefficients are zeros.

Then we applied compressive sampling with different sampling rates (α) (0.0016, 0.003, 0.006, 0.013, 0.025, 0.05 and 0.1) with 8, 16, 32, 64, 128, 256, and 512 compressed measurements of our original vibration signal using random Gaussian matrix. The size of Gaussian matrix is m by N, where N is the length of the original vibration signal measurements and m is the number of compressed signal elements (i.e., m = $\alpha * $N). Based on compressive sampling framework, multiplying this matrix with our signal sparse representations generates different sets of compressed measurements of the vibration signal.

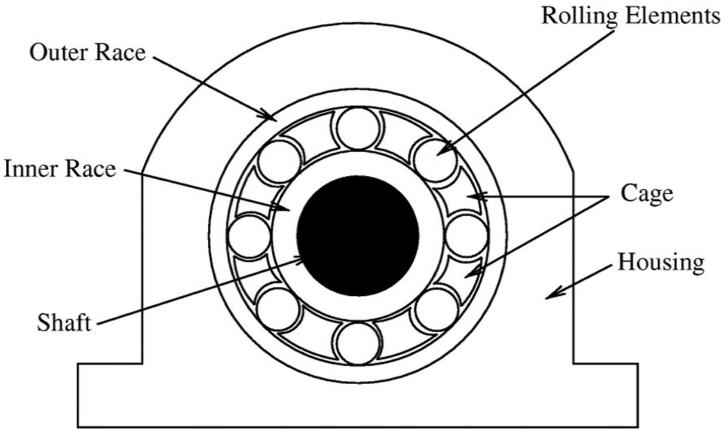

Fig. 4. A typical roller bearing.

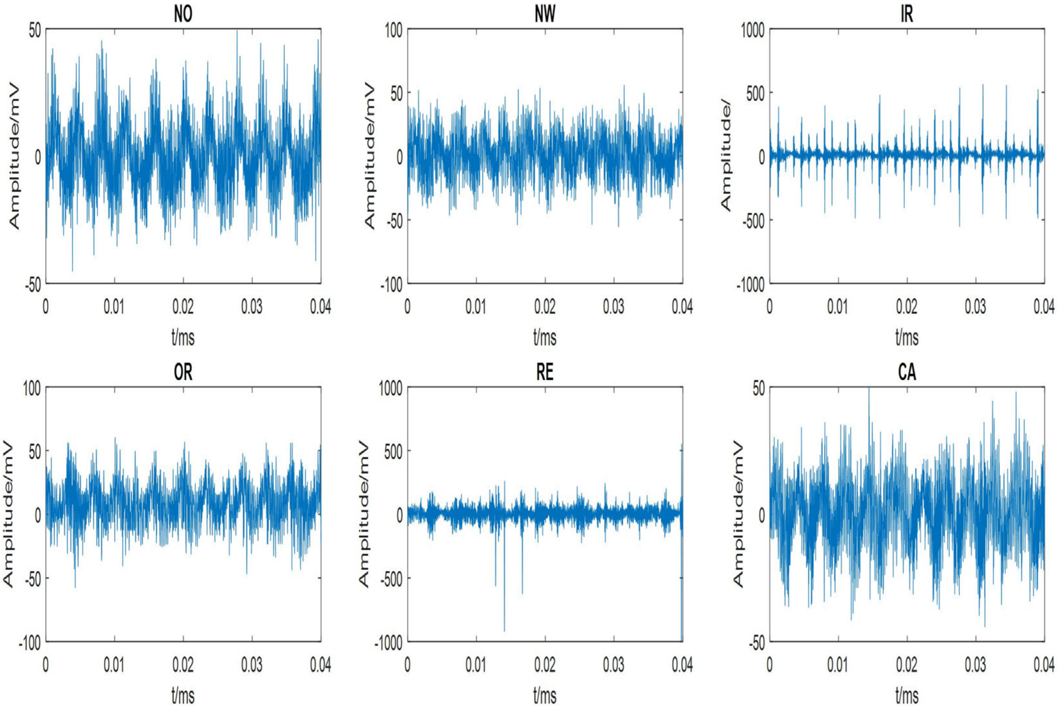

Fig. 5. Typical time – domain vibration signals for the six different conditions.

The obtained compressed measurements must possess the quality of the original signal, i.e., have sufficient information of the original signal. Thus, we need to test that our CS model generates enough samples for the purpose of bearings fault classification. Roman et al. proposed a generalized flip test [65] for CS model that has the ability to test the efficiency of any sparsity model, any signal class, any sampling operator, and any recovery algorithm towards accurate CS model. The basic idea of this test is to flip sparsity basis coefficients which represent the sparse representations and then perform a reconstruction with measurements matrix using the sampling operator and the recovery algorithm. If the sparse vector (s) is not recovered within a low tolerance, then decrease the thresholding level to obtain a sparse signal and repeat until s recovered exactly. More details of the original flip test can be found in [66,67].

Following the idea of the generalized flip test, we tried different sampling rates (0.05, 0.1, 0.15, and 0.2) and tested the efficiency of our CS model by thresholding the wavelet coefficients to obtain sparse signal (s) and then by reconstructing s

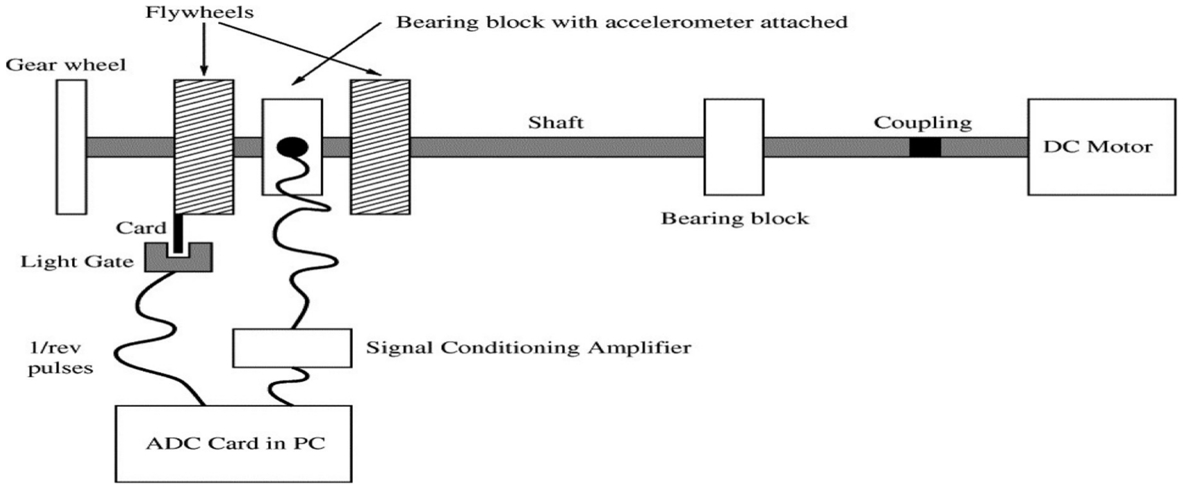

Fig. 6. The test rig used to collect the vibration data of bearings.

Table 1
Description of bearing dataset.

	Bearing conditions	Number of samples	Number of data points
Normal	NO	160	5120
	NW	160	5120
Fault	IR	160	5120
	OR	160	5120
	RE	160	5120
	CA	160	5120

from the obtained compressed measurements using random Gaussian matrix. The compressive sampling matching pursuit (CoSaMP) algorithm [68] is used to reconstruct the sparse signal.

The reconstruction errors measured by Root Mean Squared Error (RMSE) for the six conditions of bearings are presented in the following Table 2. The second column depicts the reconstruction errors compared to the original thresholded coefficients using 5% of the original signal for the six conditions, the third column shows the reconstructions errors using 10% of the original signal, the fourth is for 15% of the original signal and the fifth column is for 20% of the original signal.

It is clear that as α increases RMSE decreases, indicating better signal reconstruction. The better signal reconstruction indicate that the compressed measurements possess the quality of the original signal.

5.1.3. Experiments

In order to verify the validity of the proposed method, we carried out several experiments to learn over-complete features of various highly compressed bearing data sets obtained by using different compressed sampling rate. Fifty percent of these compressed samples are randomly selected for the pre-training stage of DNN, then these samples are used to re-train the deep net, the other 50% of samples are used for testing the performance. Then, the obtained over-complete features used for classification using different settings of DNN. Finally, we compare our proposed method, using our highly compressed datasets, with several existing methods.

To learn features from these compressed measurements, we used a sparse autoencoder neural network with a limited number of hidden layers (2, 3, and 4 hidden layers). The structures of these different hidden layers are chosen to be in the form of over-complete feature learning (expansion), where the number of neurons in different hidden layers network structure is twice the number of neurons in preceding layer, for example, if the number of input samples in the input layer is z then the number of nodes in the first hidden layer is 2z and 4z in the second hidden layer and so on. The number of nodes of the output layer is limited by the number of bearing conditions (6 conditions). A Bi-directional deep architecture of stacked autoencoders has been used for the purpose of deep learning, these include feedforward and backpropagation (BP). The parameters that control the effects of using regularizers by sparse autoencoder were set as follows: the weight decay (λ) was set to very small value 0.002, the weight of the sparsity penalty term (β) was set to be 4, and the sparsity parameter (ρ) to 0.1. The maximum training epoch is 200.

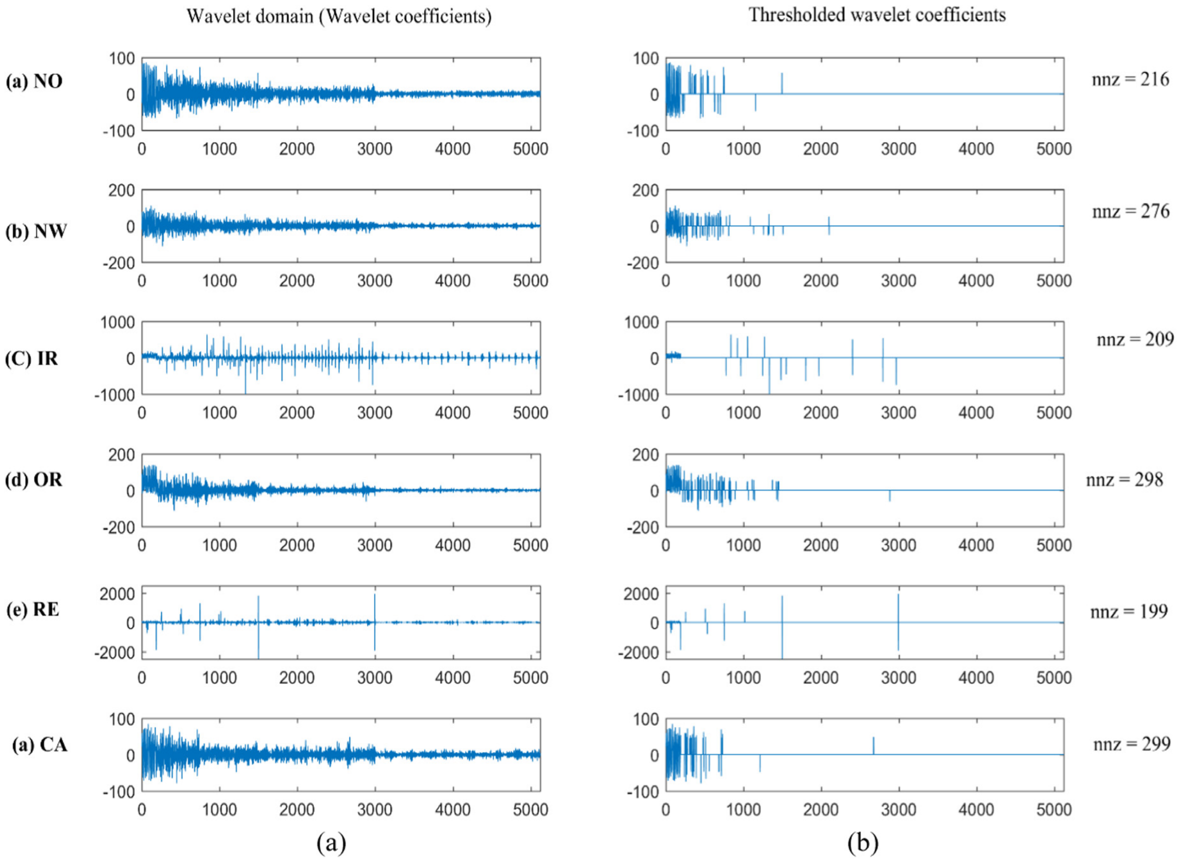

Fig. 7. Wavelet coefficients and corresponding thresholded wavelet coefficients for each condition signal (nnz refers to number of non-zero elements).

Table 2
Results of root-mean-square-error (RMSE) for various sampling rates.

	$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$	$\alpha = 0.20$
NO	9.37	8.45	0.04	0.03
NW	11.29	24.64	5.8	0.06
IR	20.82	15.23	0.16	0.09
OR	14.38	12.71	8.18	0.07
RE	17.02	11.87	0.35	0.12
CA	8.3	5.29	0.14	0.04

5.1.4. Results

The previous sections have described the principle structure of our proposed method and the experimental setup. Various experiments were conducted to use our method for classifying bearing faults from different highly compressed vibration measurements. The overall classification results obtained from these experiments are shown in Table 3. The first major comment is that the classification accuracy after the second stage is better than that after the first stage for every dataset at these values of α. The classification in the deep net stage (the first stage) achieved good results for larger numbers of measurements, i.e., for values of m equal to 512, 256 and 128, and high accuracy was achieved by the two hidden layers DNN using only 64 samples of our signal. Most of the classification accuracies for the two, three and four hidden layers DNNs using fine-tuning stage (the second stage) are 99% or above and some are 100% for even less than 1% compressed measurements of the original vibration signal, i.e., when $\alpha = 0.006$. The two hidden layers DNN achieved high classification accuracy (98%) for α equal to 0.003 and 0.0016 with 16 and 8 compressed measurements. Moreover, the three hidden layers DNN achieved 100% with only 16 measurements, i.e., $\alpha = 0.003$. Taken together, these results show that the proposed method has the ability to classify the bearing conditions with a high accuracy from highly compressed bearing vibration measurements.

In comparison with DNN based on sparse autoencoder using under-complete representations, i.e., when the number of nodes in each hidden layer is less than the number of input samples. Classification results from several experiments using

Table 3
Overall classification accuracies and their related standard deviations.

Compressive sampling rate								
Number of hidden layers	Classification stage	α = 0.0016 (m = 8)	α = 0.003 (m = 16)	α = 0.006 (m = 32)	α = 0.013 (m = 64)	α = 0.025 (m = 128)	α = 0.05 (m = 256)	α = 0.1 (m = 512)
2	Deep net (the first stage)	95.3 ± 1.0	95.9 ± 0.3	96.0 ± 1.3	99.1 ± 0.3	99.8 ± 0.1	100	99.8 ± 0.1
	Fine – tuning (the second stage)	98.0 ± 0.3	98.8 ± 0.3	99.6 ± 0.1	99.6 ± 0.1	99.8 ± 0.1	100	99.8 ± 0.1
3	Deep net (the first stage)	64.8 ± 4.2	94.9 ± 2.4	85.6 ± 4.4	93.6 ± 2.2	99.2 ± 2.6	99.6 ± 0.1	100
	Fine – tuning (the second stage)	82.0 ± 5.1	100	99.5 ± 0.3	99.8 ± 0.1	100	100	100
4	Deep net (the first stage)	36.6 ± 8.3	55.2 ± 6.1	90.5 ± 2.9	95.6 ± 1.2	99.8 ± 0.1	98.6 ± 3.9	100
	Fine – tuning (the second stage)	83.1 ± 1.6	89.4 ± 2.5	96.3 ± 0.8	99.7 ± 0.3	99.8 ± 0.1	99.7 ± 0.3	100

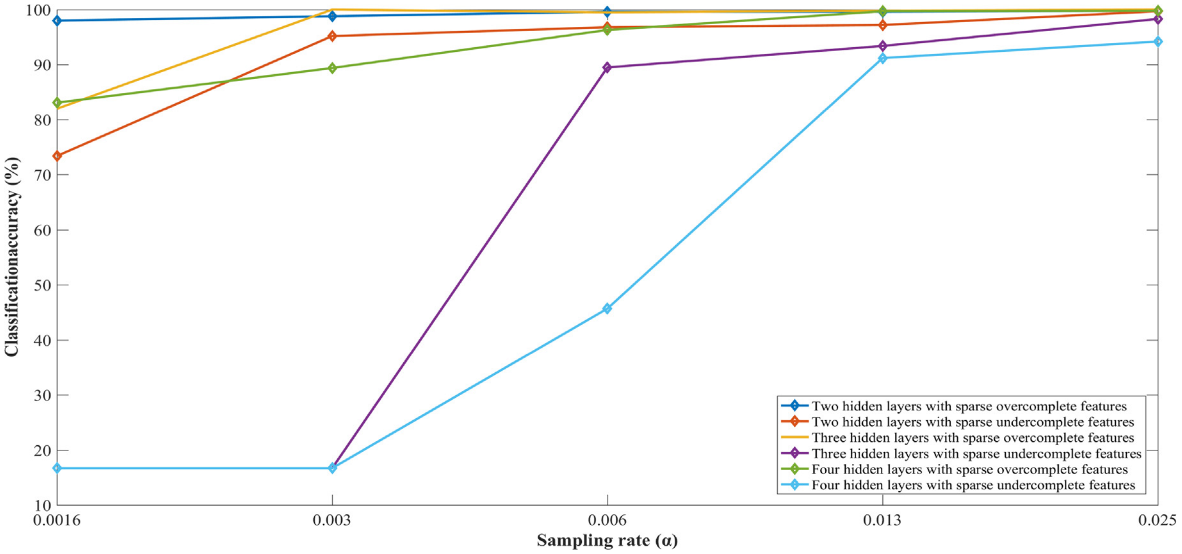

Fig. 8. Classification performance of under-complete and over-complete feature representations with two, three and four hidden layers DNN.

under-complete feature representations that also deal with the same highly compressed datasets are compared to the results acquired using over-complete representations as shown in Fig 8. From this figure, it can be clearly seen that all the scenarios of DNNs that used over-complete feature representations outperform all those utilizing under-complete sparse features when the input samples are extremely compressed datasets (i.e., with only 8 and 16 compressed measurements). Evidently, the two hidden layers DNNs in all scenarios achieved better results than other network structures, i.e., three hidden layers DNN and four hidden layers DNN.

In order to measure the training performance of the proposed over-complete feature based DNN compared to under-complete representations based DNN, a typical value of α = 0.025 and two hidden layers DNN has been used in this comparison. Fig 9 shows that minimum Mean Squared Error (MSE) value which is 0.003 was achieved with the training of three hidden layers DNN using over-complete features in epoch 200, compared to the same DNN structure using under-complete feature representations where MSE was 0.021.

For further verification of the performance of the proposed method, complete comparison results of the classification accuracy using the proposed technique compared with three classifiers, namely, logistic regression classifier (LRC), support vector machine (SVM) and neural network (NN) were used to classify faults from the same highly compressed measurements sets and the complete results are shown in Table 4. It is clear that results from our proposed method with smaller sampling rate as in α = 0.0016, 0.003, and 0.006 are better than those achieved by other classifiers.

5.1.4.1. Effects of parameterization on the classification accuracy. To control the effects of parameterization on SAEs, some parameters need to be set, these include, sparsity parameter (ρ), weight decay (λ) and the weight of the sparsity penalty term (β). To test the influence of these parameters values on bearing fault classification performance, several experiments have been carried out using our proposed method with two hidden layers, and different values of SAE parameterization. The sampling rate α was set to 0.05 where two hidden layers achieved 100% classification accuracy for both classification stages, i.e.,

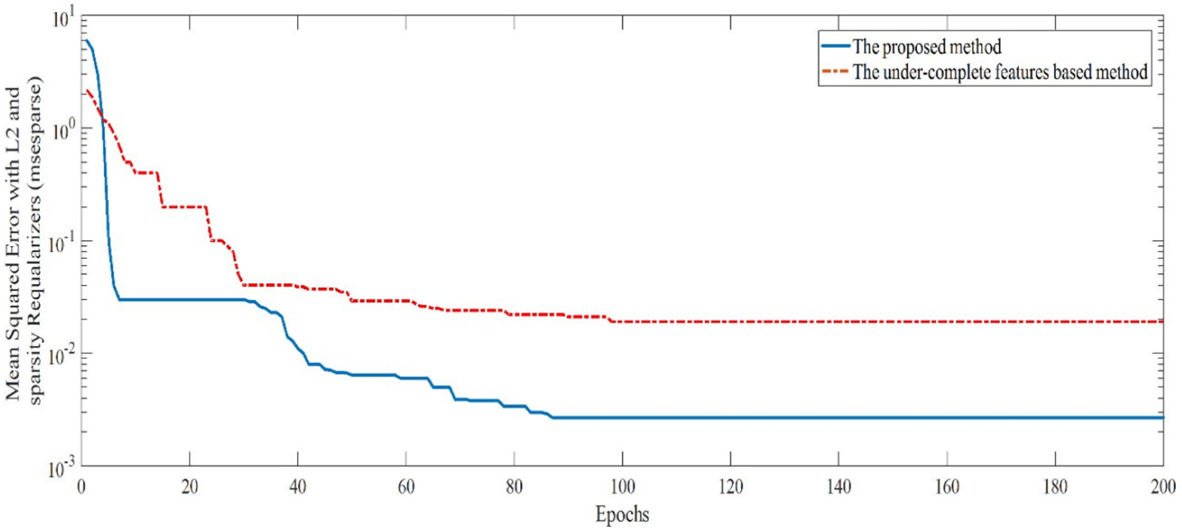

Fig. 9. Training performance of over-complete feature based two hidden layers DNN and under complete feature based two hidden layers DNN (α = 0.025).

Table 4
Complete classification results and their related standard deviations using LRC, SVM, NN and the proposed method.

	Compressive sampling rate						
Classification method	α = 0.0016 (m = 8)	α = 0.003 (m = 16)	α = 0.006 (m = 32)	α = 0.013 (m = 64)	α = 0.025 (m = 128)	α = 0.05 (m = 256)	α = 0.1 (m = 512)
LRC	64.2 ± 7.9	92.2 ± 5.5	96.2 ± 0.6	98.2 ± 0.5	99.0 ± 0.5	99.0 ± 0.3	99.3 ± 0.3
SVM	77.7 ± 19.3	94.3 ± 2.3	97.8 ± 0.4	98.4 ± 0.4	99.5 ± 0.3	99.7 ± 0.2	99.7 ± 0.2
ANN	60.4 ± 10.5	76.7 ± 6.2	98.0 ± 1.9	98.8 ± 0.3	99.6 ± 0.2	99.7 ± 0.2	99.7 ± 0.2
Proposed method	98.0 ± 0.3	98.8 ± 0.3	99.6 ± 0.1	99.6 ± 0.1	99.8 ± 0.1	100	99.8 ± 0.1

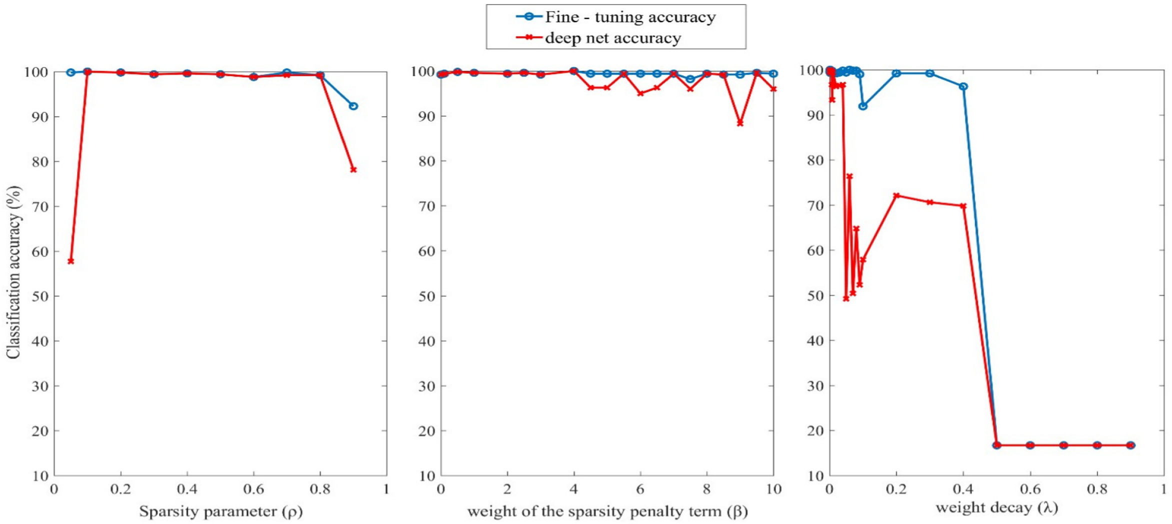

Fig. 10. Effects of parameterization on the classification accuracy.

deep net (the first stage) and fine-tuning (the second stage) with the parameters values described in section 4.1.3. As can be seen from Fig 10, while the classification accuracy is sensitive to the value of λ, there are very broad range of values for ρ and β for which classification accuracies are very high and stable.

5.1.5. Comparison of results

In this subsection, a comparison of several methods using the same vibration dataset as in [32]. One method uses all the original samples. Each of the two other methods uses compressed measurements (for values of α of 0.5 and 0.25) and then reconstruct the original signals. These three have been reported in [32]. The remaining three are our proposed methods to demonstrate the possibility to sample the vibration data of roller element bearings at less than Nyquist rate using CS and to perform fault classification without reconstructing the original signal. Table 5 shows classification results of bearing faults using our proposed method with two hidden layers and a sampling rate α of 0.5, 0.25 and 0.1 and the reported results in [32] using the same dataset. It is clear that all our results are better than those in [32].

5.2. Second case study

5.2.1. Data description

The bearing datasets used in this case are provided by the Case Western Reserve University [69]. The data were acquired from a motor driving mechanical system where the faults were seeded into the drive-end bearing of the motor. The bearing datasets were collected under normal condition (NO), with inner race fault (IR), with roller element fault (RE) and with outer race fault (OR). The datasets are further categorised by fault width (0.18–0.53 mm) and motor load (0–3 hp). The sampling rate used were 48 kHz, a good description of the test setup can be found in [69,70].

In this study, three categories of bearings datasets, i.e., A, B and C based on motor load were used to test the performance of the proposed method. The description of these datasets is presented in Tables 6 and 7. Under motor loads of 1, 2 and 3 hp. These datasets contain 10 bearing health conditions and have 200 samples for each condition, and each signal contains 2400 data points.

We applied the same data processing steps as in case I to each dataset (i.e., A, B and C) to obtain compressed vibration signals with different sampling rates (α) (as 0.025, 0.05 and 0.1, and 0.2) with 60, 120, 240 and 480 compressed measure-

Table 5
A comparison with the classification results from literature on bearing dataset.

		Accuracy (%)
Raw vibration [32]		98.9 ± 1.2
Compressed sensed (α = 0.5) [32] followed by reconstruction		92.4 ± 0.5
Compressed sensed (α = 0.25) [32] followed by reconstruction		84.6 ± 3.4
This work (α = 0.5)	Deep net (the first stage)	99.1 ± 1.7
(two hidden layers)	Fine – tuning (the second stage)	100 ± 0.0
This work (α = 0.25)	Deep net (the first stage)	99.6 ± 1.2
(two hidden layers)	Fine – tuning (the second stage)	100 ± 0.0
This work (α = 0.1)	Deep net (the first stage)	99.8 ± 0.1
(two hidden layers)	Fine – tuning (the second stage)	99.8 ± 0.1

Table 6
Description of the three bearing datasets.

Datasets	Load	Number of samples in each condition
A	1	200
B	2	200
C	3	200

Table 7
Description of the bearing health conditions.

Fault type	Fault width in (mm)	Classification label
NO	0	1
IR1	0.18	2
IR2	0.36	3
IR3	0.53	4
RE1	0.18	5
RE2	0.36	6
RE3	0.53	7
OR1	0.18	8
OR2	0.36	9
OR3	0.53	10

ments of A, B and C original vibration signals. Fifty percent of these compressed samples are randomly selected for the pre-training stage of DNN, then these samples are used to re-train the deep net, the other 50% of samples are used for testing the performance.

5.2.2. Classification results

The proposed method with two hidden layers is used to process the compressed measurements of each dataset. The classification accuracy rates are obtained by averaging the results of ten experiments for each compressed datasets obtained using the different sampling rates described in section 4.2.1; 50% of samples are selected for training and the other samples are used for testing. The average accuracies and their corresponding standard deviations of ten experiments for each dataset are shown in Table 8. One of the more significant findings to emerge from the results in Table 8 is that classification results after the fine-tuning stage (the second stage) is better than that after the first stage for all datasets A, B and C with different values of α. Also, it shows that the deep net stage (the first stage) achieved good results with 99.6% and 99.5% for $\alpha = 0.2$ with datasets B and C respectively. Most of the classification accuracy results after the second stage are above 99% for values of α in the range of 0.05 to 0.2. In particular, results after the second stage of our proposed method for datasets B and C with α equal to 0.2 achieved 100% accuracy for every single run in our investigations, and also the 100% accuracy achieved for data C for α equal to 0.1. Overall, these results indicate that the proposed method has the ability to classify the bearing conditions with high accuracy from highly compressed vibration measurements.

5.2.3. Comparison of results

To evaluate the effectiveness of our proposed method, Table 9 presents the comparisons with some recently published results [52,71] with the same roller bearing datasets A, B and C. The second left hand column presents the classification results of DNN based method in [52] while the third column shows the classification results of back propagation neural network (BPNN) based method in [52]. In [71] a generic multi-layer perceptron (MLP) was used for the classification purpose.

It is clear that results from our proposed methods with fine-tuning (second stage) are very competitive. In particular, results from our fine-tuning method with dataset C achieved 100% accuracy for every single run in our investigations, even

Table 8
Classification results for CWRU datasets.

Compressive sampling rate					
Datasets	Classification stage	$\alpha = 0.025$ (m = 60)	$\alpha = 0.05$ (m = 120)	$\alpha = 0.1$ (m = 240)	$\alpha = 0.2$ (m = 480)
A	Deep net (the first stage)	89.7 ± 5.4	93.7 ± 2.9	92.5 ± 2.2	98.7 ± 0.8
	Fine – tuning (the second stage)	97.6 ± 0.7	98.4 ± 1.3	99.3 ± 0.6	99.8 ± 0.2
B	Deep net (the first stage)	94.5 ± 1.1	96.2 ± 1.6	98.2 ± 0.6	99.6 ± 0.6
	Fine – tuning (the second stage)	98.9 ± 0.2	99.3 ± 0.4	99.7 ± 0.5	100 ± 0.0
C	Deep net (the first stage)	97.3 ± 0.8	98.5 ± 1.4	98.2 ± 0.4	99.5 ± 0.2
	Fine – tuning (the second stage)	99.4 ± 0.6	99.7 ± 0.3	100 ± 0.0	100 ± 0.0

Table 9
A comparison with the results from literature on CWRU vibration datasets of roller bearings.

Method	[52]	[52]	[71]	Our proposed method with fine-tuning
Dataset	DNN	BPNN	MLP	(the second stage) with 2 hidden layers $\alpha = 0.1$
A	99.95 ± 0.06	65.20 ± 18.09	95.7	99.3 ± 0.6
B	99.61 ± 0.21	61.95 ± 22.09	99.6	99.7 ± 0.5
C	99.74 ± 0.16	69.82 ± 17.67	99.4	100 ± 0.0

Table 10
A comparison results to examine the speed and accuracy performances in several scenarios.

Dataset	WT-DNN with fine-tuning				Our proposed method with fine-tuning (the second stage)	
	Using DNN implementation of [52] with 2 hidden layers (with no CS)		Using DNN implementation in this paper with 2 hidden layers (with no CS)		WT - CS- DNN (2 hidden layers)	
	Accuracy (%)	Time (mins)	Accuracy (%)	Time (mins)	Accuracy (%)	Time (mins)
A	99.0 ± 0.1	41.5 ± 0.3	99.4 ± 0.5	34.1 ± 0.7	99.3 ± 0.6	5.7 ± 0.1
B	99.1 ± 0.6	43.3 ± 0.7	99.5 ± 0.8	32.9 ± 0.3	99.7 ± 0.5	5.9 ± 0.4
C	99.6 ± 0.3	43.1 ± 0.2	99.8 ± 1.1	34.2 ± 0.9	100	5.7 ± 0.2

though we are using a limited amount (only 10%) of the original data, which are not matched by any of the other methods using 100% of the data.

For further verification of the efficiency of the proposed method, we conducted three experiments (all with 2 hidden layers and fine-tuning) to examine the speed and accuracy performances in several scenarios. The results are presented in Table 10. The first column refers to the three datasets. The second and third columns describe accuracies and execution times of using a "traditional" autoencoder based DNN of [52] with 2400 inputs from Haar Wavelet (with no CS), while the fourth and fifth columns describe accuracies and execution times of using our sparse autoencoder based DNN with 2400 inputs from Haar Wavelet (with no CS). Two things are clear for each of the three datasets – 1) our autoencoder based DNN (even without CS) is much faster than (or requires only 80% of the time of) the "traditional" autoencoder based DNN, and 2) yet our classification results are very competitive.

The sixth and seventh columns describe accuracies and execution times of using our proposed sparse autoencoder based DNN with 240 inputs from Haar Wavelet (with CS). Two things are clear for each of the three datasets – (1) our autoencoder based DNN (even with CS) is significantly faster than (or requires only 15% of the time of) the "traditional" autoencoder based DNN, and (2) yet our classification results are as good as, if not better than, the other two scenarios.

In summary, the significant reduction in computation time comes from two sources – (1) using our proposed sparse autoencoder and (2) using CS. Finally, our complete proposal (using sparse autoencoder and CS) achieves classification results for all three datasets that are as good as, if not better than, the other two scenarios.

6. Conclusion

In this investigation, the aim was to assess classification of bearing faults from highly compressed measurements based on CS. The proposed method includes the extraction of over-complete sparse representations from highly compressed measurements. It involves the unsupervised feature learning algorithm SAE for learning feature representations in multi-stages of non-linear feature transformation based on DNN. The accuracy of the proposed method is verified using highly compressed datasets of rolling element bearings signals obtained using different compressed sampling rates. These compressed datasets contain fewer samples for each bearing condition. The most obvious finding to emerge from this study is that, despite achieving fairly high classification accuracy in the first stage, the proposed method is able to achieve higher classification accuracy in the second stage even from highly compressed measurements compared to the existing methods. Moreover, classification results from our proposed method outperform those achieved by reconstructing the original signals. Additionally, a significant reduction in computation time is achieved using our proposed method compared to another autoencoder based DNN method [52], with better classification accuracies. The implication of this is the possibility that the proposed method of compressive sensing in machine faults classification will require fewer measurements thus it would reduce the computational complexity, storage requirement and the bandwidth for transmitting reduced data.

Acknowledgement

Professor Nandi is a Distinguished Visiting Professor at Tongji University, Shanghai, China. This work was partly supported by the National Science Foundation of China [grant number 61520106006], and the National Science Foundation of Shanghai [grant number 16JC1401300].

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ymssp.2017.06.027.

References

[1] R.B. Randall, J. Antoni, Rolling element bearing diagnostics – a tutorial, Mech. Syst. Signal Process. 25 (2011) 485–520.

[2] T.H. Loutas, D. Roulias, E. Pauly, V. Kostopoulos, The combined use of vibration, acoustic emission and oil debris on-line monitoring towards a more effective condition monitoring of rotating machinery, Mech. Syst. Signal Process. 25 (4) (2011) 1339–1352.

[3] A.K. Nandi, C. Liu, M.L.D. Wong, Intelligent vibration signal processing for condition monitoring, Surveill. 7 Int. Conf. – Oct. 29-30, 2013 Inst. Technol. Chartres, Fr. Plenary Session, 2013.

[4] L. Zhang, A.K. Nandi, Fault classification using genetic programming, Mech. Syst. Signal Process. 21 (3) (2007) 1273–1284.

[5] M.L.D. Wong, L.B. Jack, A.K. Nandi, Modified self-organising map for automated novelty detection applied to vibration signal monitoring, Mech. Syst. Signal Process. 20 (3) (2006) 593–610.

[6] A.K.S. Jardine, D. Lin, D. Banjevic, a review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (2006) 1483–1510.

[7] Michael Peter Norton, Denis G. Karczub, Fundamentals of noise and vibration analysis for engineers, Cambridge University Press, 2003.

[8] N.G. Nikolaou, I.A. Antoiadis, Demodulation of vibration signals generated by defects in rolling element bearings using complex shifted morlet wavelets, Mech. Syst. Signal Process. 16 (2002) 677–694.

[9] D.H. Pandya, S.H. Upadhyay, S.P. Harsha, Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform, Soft. Comput. 18 (2014) 255–266.

[10] D.H. Pandya, S.H. Upadhyay, S.P. Harsha, Ann based fault diagnosis of rolling element bearing using time-frequency domain feature, J. Eng. Sci. Technol. 4 (2012) 2878–2886.

[11] Y.G. Lei, J. Lin, Z.J. He, et al, A review on empirical mode decomposition in fault diagnosis of rotating machinery, Mech. Syst. Signal Process. 35 (1–2) (2013) 108–126.

[12] B.A. Paya, I.I. Esat, M.N.M. Badi, Artificial neural network based fault diagnostics of rotating machinery using wavelet transforms as a pre-processor, Mech. Syst. Signal Process. 11 (1997) 751–765.

[13] C. Wang, M. Gan, C. Zhu, Non-negative EMD manifold for feature extraction in machinery fault diagnosis, Meas. J. Int. Meas. Confed. 70 (2015) 188–202.

[14] Z. Feng, M. Liang, F. Chu, Recent advances in time-frequency analysis methods for machinery fault diagnosis: a review with application examples, Mech. Syst. Signal Process. 38 (2013) 165–205.

[15] R. Yan, R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: a review with applications, Signal Process. 96 (2014) 1–15.

[16] Jérôme Antoni, R.B. Randall, The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines, Mech. Syst. Signal Process. 20 (2) (2006) 308–331.

[17] B. Eftekharnejad, M.R. Carrasco, B. Charnley, D. Mba, The application of spectral kurtosis on acoustic emission and vibrations from a defective bearing, Mech. Syst. Signal Process. 25 (1) (2011) 266–284.

[18] I. Antoniadis, G. Glossiotis, Cyclostationary analysis of rolling-element bearing vibration signals, J. Sound Vib. 248 (5) (2001) 829–845.

[19] A.C. McCormick, A.K. Nandi, Cyclostationarity in rotating machine vibrations, Mech. Syst. Signal Process. 12 (2) (1998) 225–242.

[20] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (2006) 1289–1306.

[21] E.J. Candes, M.B. Wakin, An Introduction to Compressive Sampling, IEEE Signal Process. Mag. 25 (2008).

[22] M. Rossi, A.M. Haimovich, Y.C. Eldar, Spatial compressive sensing for MIMO radar, IEEE Trans. Signal Process. 62 (2014) 419–430.

[23] S. Li, L. Da Xu, X. Wang, Compressed sensing signal and data acquisition in wireless sensor networks and internet of things, IEEE Trans. Ind. Infor. 9 (2013) 2177–2186.

[24] D.J. Holland, D.M. Malioutov, A. Blake, A.J. Sederman, L.F. Gladden, reducing data acquisition times in phase-encoded velocity imaging using compressed sensing, J. Magn. Reson. 203 (2010) 236–246.

[25] S.L. Merlet, R. Deriche, Continuous diffusion signal, EAP and ODF estimation via compressive sensing in diffusion MRI, Med. Image Anal. 17 (2013) 556–572.

[26] S. Qaisar, R.M. Bilal, W. Iqbal, M. Naureen, S. Lee, Compressive sensing: from theory to applications, a survey, J. Commun. Networks. 15 (2013) 443–456.

[27] C.R. Berger, Z. Wang, J. Huang, S. Zhou, Application of compressive sensing to sparse channel estimation, IEEE Commun. Mag. 48 (2010) 164–174.

[28] F. Renna, R. Calderbank, L. Carin, M.R.D. Rodrigues, Reconstruction of signals drawn from a Gaussian mixture via noisy compressive measurements, IEEE Trans. Signal Process. 62 (2014) 2265–2277.

[29] W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, IEEE Trans. Inf. Theory 55 (2009) 2230–2249.

[30] E.J. Candès, Y.C. Eldar, D. Needell, P. Randall, Compressed sensing with coherent and redundant dictionaries, Appl. Comput. Harmon. Anal. 31 (2011) 59–73.

[31] S. Stanković, I. Orović, L. Stanković, an automated signal reconstruction method based on analysis of compressive sensed signals in noisy environment, Signal Process. 104 (2014) 43–50.

[32] M.L.D. Wong, M. Zhang, A.K. Nandi, Effects of compressed sensing on classification of bearing faults with entropic features, in: EUSIPCO Proceeding, IEEE, 2015, pp. 2296–2300.

[33] X.F. Li, X.C. Fan, L.M. Jia, Compressed sensing technology applied to fault diagnosis of train rolling bearing, Vib. Struct. Eng. Meas. Ii, Pts 1-3 226–228 (2012) 2056–2061.

[34] M.a. Davenport, M.B. Wakin, R.G. Baraniuk, Detection and estimation with compressive measurements, Comput. Eng (2007) 1–16.

[35] G. Tang, Q. Yang, H.Q. Wang, G. gang Luo, J. wei Ma, Sparse classification of rotating machinery faults based on compressive sensing strategy, Mechatronics 31 (2014) 60–67.

[36] X. Zhang, N. Hu, L. Hu, L. Chen, Z. Cheng, A bearing fault diagnosis method based on the low-dimensional compressed vibration signal, Adv. Mech. Eng. 7 (2015) 1–12.

[37] X. Chen, Z. Du, J. Li, X. Li, H. Zhang, Compressed sensing based on dictionary learning for extracting impulse components, Signal Process. 96 (2014) 94–109.

[38] G. Tang, W. Hou, H. Wang, G. Luo, J. Ma, Compressive sensing of roller bearing faults via harmonic detection from under-sampled vibration signals, Sensors (Switzerland) 15 (2015) 25648–25662.

[39] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, X. Gao, Deep Learning and its applications to machine health monitoring: a survey, 2016. .

[40] L. Ren, W. Lv, S. Jiang, Y. Xiao, Fault diagnosis using a joint model based on sparse representation and SVM, IEEE Trans. Instrum. Meas. (2016) 1–8.

[41] H. Zhu, X. Wang, Y. Zhao, Y. Li, W. Wang, L. Li, Sparse representation based on adaptive multiscale features for robust machinery fault diagnosis, Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci. 229 (2015) 2303–2313.

[42] H. Tang, J. Chen, G. Dong, Sparse representation based latent components analysis for machinery weak fault detection, Mech. Syst. Signal Process. 46 (2014) 373–388.

[43] H. Liu, C. Liu, Y. Huang, Adaptive feature extraction using sparse coding for machinery fault diagnosis, Mech. Syst. Signal Process. 25 (2011) 558–574.

[44] W. Fan, G. Cai, Z.K. Zhu, C. Shen, W. Huang, L. Shang, Sparse representation of transients in wavelet basis and its application in gearbox fault feature extraction, Mech. Syst. Signal Process. 56 (2015) 230–245.

[45] M.S. Lewicki, T.J. Sejnowski, Learning overcomplete representations, Neural Comput. 12 (2000) 337–365, http://dx.doi.org/10.1162/089976600300015826.

[46] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: A strategy employed by V1?, Vision Res. 37 (1997) 3311–3325.

[47] E. Doi, D. Balcan, A theoretical analysis of robust coding over noisy overcomplete channels, Adv. Neural. Inf. (2006), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.9617&rep=rep1&type=pdf.

[48] A. Ng, CS294A lecture notes sparse autoencoder, Cs294a (2011) 1–19.

[49] H. Lee, C. Ekanadham, A. Ng, Sparse deep belief net model for visual area V2, Nips (2007) 1–8.

[50] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2007) 153.

[51] S. Tao, T. Zhang, J. Yang, X. Wang, W. Lu, Bearing fault diagnosis method based on stacked autoencoder and softmax regression, in: Chinese Control Conf. CCC. 2015-September; 2015, pp. 6331–6335.

[52] F. Jia, Y. Lei, J. Lin, X. Zhou, N. Lu, Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, Mech. Syst. Signal Process. 72–73 (2016) 303–315.

[53] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, X. Chen, A sparse auto-encoder-based deep neural network approach for induction motor faults classification, Measurement 89 (2016) 171–178.

[54] E. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies?, IEEE Trans Inf. Theory 52 (12) (2006) 5406–5425.

[55] Richard Baraniuk et al, A simple proof of the restricted isometry property for random matrices, Constr. Approx. 28 (3) (2008) 253–263.

[56] J. Schmidhuber, Deep Learning in neural networks: an overview, Neural Networks 61 (2015) 85–117.

[57] D. Erhan, A. Courville, P. Vincent, Why does unsupervised pre-training help deep learning, J. Mach. Learn. Res. 11 (2010) 625–660.

[58] S. Lange, M. Riedmiller, Deep auto-encoder neural networks in reinforcement learning. In: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE; 2010 Jul 18, pp. 1–8.

[59] Tim Salimans, Diederik P. Kingma, Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks; 2016. .

[60] Hosmer Jr, W. David, Stanley Lemeshow, Rodney X. Sturdivant, Applied Logistic Regression, John Wiley & Sons, 2013.

[61] Martin Fodslette Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Netw. 6 (4) (1993) 525–533.

[62] Yuequan Bao, James L. Beck, Hui Li, Compressive sampling for accelerometer signals in structural health monitoring, Struct. Health Monit. (2010).

[63] Yuequan Bao, Hui Li, Ou. Jinping, Emerging data technology in structural health monitoring: compressive sensing technology, J. Civil Struct. Health Mon. 4 (2) (2014) 77–90.

[64] http://www.ece.northwestern.edu/local-apps/matlabhelp/toolbox/wavelet/ch06_a47.html.

[65] B. Roman, A. Bastounis, B. Adcock, A.C. Hansen, On Fundamentals of Models and Sampling in Compressed Sensing, Available: <http://www.damtp.cam.ac.uk/research/afha/bogdan/ Fundamentals_17.pdf>

[66] B. Adcock, A.C. Hansen, C. Poon, B. Roman, Breaking the Coherence Barrier: A New Theory for Compressed Sensing, 2014. .

[67] B. Roman, B. Adcock, A. Hansen, On Asymptotic Structure in Compressed Sensing, 2014. .

[68] D. Needell, J.A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, Appl. Comput. Harmon. Anal. 26 (3) (2009) 301–321.

[69] Case Western Reserve University Bearing Data Center <http://csegroup.case.edu/bearingdatacenter/home>

[70] Wade A. Smith, Robert B. Randall, Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark study, Mech. Syst. Signal Process. 64 (2015) 100–131.

[71] L.F. de Almeida, J.W. Bizarria, F.C. Bizarria, M.H. Mathias, Condition-based monitoring system for rolling element bearing using a generic multi-layer perceptron, J. Vib. Control 21 (16) (2015) 3456–3464.