Complex Intell. Syst. (2018) 4:67–77
https://doi.org/10.1007/s40747-017-0054-8

ORIGINAL ARTICLE

Restricted Boltzmann machine and softmax regression for fault detection and classification

Praveen Chopra[1] · **Sandeep Kumar Yadav**[2]

Received: 18 April 2016 / Accepted: 25 July 2017 / Published online: 5 August 2017

Abstract A unique technique is proposed based on restricted Boltzmann machine (RBM) and softmax regression for automated fault detection and classification using the acoustic signal generated from IC (Internal Combustion) engines. This technique uses RBM for unsupervised fault feature extraction from the frequency spectrum of the noisy acoustic signal. These extracted features are then used to reduce the dimensionality of the training and testing data vectors. These reduced dimensionality data vectors are used by softmax regression-based classifier for classification of the engine into faulty and healthy class. The proposed technique does not require any hand-engineered feature extraction, as usually done. This technique performs very well with a small number of training data. The overall performance of this technique for four different fault classes is more than 99% on the industrial IC engine data. In a typical case, with only 38 training data sets and 210 test data sets, the performance is 99.52%.

Keywords Restricted Boltzmann machine · Softmax regression · Feature extraction · Fault detection

Introduction

In the literature, most of the automatic fault detection uses the acoustic or vibration signal generated by the engines for fault detection and classification, as most of the mechanical faults have noticeable indicators in the form of vibration and acoustic signals [1]. However, it is quite difficult to handle these huge sizes of time-domain signal data for feature extraction. Due to non-stationary and highly dynamic nature of these vibration or acoustic signals, in the literature, most of the techniques developed so far are transforming the time-domain data into frequency-domain or time–frequency domain. This way the time-domain data are represented in a small size of vector or matrix and further used for feature extraction.

In time–frequency domain, the most widely used digital signal processing technique is the wavelet-packet transform (WPT). Yen and Lin [2] have proposed WPT-based feature extraction technique from vibration data. In this technique, they have used the wavelet coefficients as the features of the vibration data. Wu [3] also proposed WPT-based feature extraction, where energy distribution of the wavelet packets is used as the features of the acoustic signal. In this work, different levels of wavelet packet decomposition with various types of mother wavelets are used to get different types of feature spaces to train ANN-based classifier. In the frequency-domain transformation, Yadav [4] has used spectrogram of the signal to extract nine statical features such as kurtosis, shape factor, crest factor, mean, median, and variance.

These feature extraction techniques, such as energy of WPT packet, are based on some hand-engineered criteria and the extracted feature space by these techniques has very large dimension to be used by a classifier. Therefore, a suitable rule or criteria are needed to reduce the dimensionality of the feature space or to select some of the features that best represents the whole feature space. These constraints restrict these techniques to be used for all types of cases.

In this technique, the FFT is used to transform the time-domain signals into its frequency spectrum. The frequency

✉ Praveen Chopra
praveenchopra@gmail.com

Sandeep Kumar Yadav
sy@iitj.ac.in

[1] DRDO, Delhi, India

[2] Indian Institute of Technology Jodhpur, Rajasthan, India

spectrum represents the large size of time-domain data in a small size vector and removes the repetition of the data features. By this representation, the time-domain information is lost, but it does not affect the performance of the technique. In general, due to the faults, there are peaks in the spectrum at the harmonics of the operating frequency of the engine. The relation of these peak values with harmonics represents the features of the fault signal. By analyzing the fault signals, it has been observed that most of the spectrum peaks are at frequency less the 5 KHz, so the spectrum data only up to 6 KHz are used. This spectrum vector of 6 KHz frequency components is then used for feature extraction.

The main motivation behind this proposed technique is to improve the classification performance with the significantly reduced requirement of the labeled training data and training time. The other motivations to this technique are to reduce the dimensionality of the feature space to speed up the classification time and performance. The proposed technique does not require any noise filtering on the acoustic data recorded in the industrial environment.

The proposed technique uses the restricted Boltzmann machine (RBM) to do unsupervised feature extraction in small time from the fault spectrum data. An unlabeled data set is used to by an RBM1 to extract unlabeled features. These unlabeled features are used by another RBM2 as initial features or its initial weights. This RBM2 extracts the features from the labeled training data and the use of the unlabeled features as initial weights of the RBM2 reduces the requirement of the labeled training data considerably. The extracted features from the labeled data by RBM2 are then used to reduce the dimensionality of the testing and training data. These reduced dimensionality testing and training data are used by classifier. These reduced dimensionality data improve the classification performance and reduce the classification time.

RBMs are widely used for dimensionality reduction, feature extraction, and collaborative filtering [5]. The feature extraction by RBM is completely unsupervised and does not require any hand-engineered criteria. In the literature, RBM and its variants are widely used for feature extraction from images, text data, sound data, and others. Hilton [6] demonstrated the unsupervised feature learning from images and text by RBM. Salakhutdinov [5] has used the RBM with a large data set containing over 100 million user/movie ratings and demonstrated that the RBM and its variant are suitable for modeling tabular or count data. In areas other than images and text, Tylor [7] has demonstrated that RBM-based model can be used to efficiently capture complex non-linearities in the human motion data without sophisticated pre-processing or dimensionality reduction. Other application of RBM includes such as feature extraction for face recognition [8]. These application of RBMs in different fields was main motivation to use them for unsupervised feature extraction from large size of data.

The RBMs are stochastic neural networks and learn the features of the data in terms of the weight of the network [6]. These weights are initialized by random values for training by training data. This random initialization of the RBM weights requires many training examples and large number of iterations to achieve a global minima by its cost function (energy function). If there are few training data, then the cost function might achieve a local minima only and this makes the learned weights or features inconsistent. This inconsistency will then reduce the classification performance. To overcome this problem, these weights are initialized by some values that are closer to the features of the training data. To learn these initial features or weights, an unlabeled data set is used. A restricted Boltzmann machine RBM1 is trained on this unlabeled data set and learned features from this data sets are used as initial weights of RBM2. Then, the RBM2 with these initial features is trained with the labeled training data. The cost function of this RBM2 achieves the global minima in a small time with only few training data. This way RBM2 learns features from labeled training data quickly and with consistent features. This approach of use of unlabeled data set is also called "self-taught learning" and first proposed by Raina [16].

The unlabeled data set (unknown fault class) has unknown classification acoustic data and the labeled data set (known fault class) is from the known classification data. The unlabeled data set may not be exactly the same as training data set but from the same type of source. In most of the practical situations, it is not feasible to get a large number of engines with a particular type of fault or to seed a fault in large numbers of engines. However, it is quite easy to get engine fault recordings, for which fault labels are not known. These signal recording can be used as unlabeled data set.

The extracted features are further used to linearly transform the training and testing data. This transformation represents the training and testing data in terms of these extracted features. A softmax regression-based classifier [9–11] further used for classification of these testing and training data. The softmax regression is generalized version of the logistic regression [9–12], where the output class labels are multi-class classification instead of binary classification as done in the logistic regression. The softmax regression classifier is most suitable when the classes for classification are mutually exclusive. In this work, it was assumed that no two faults occur at the same time. In the area of machine learning, the softmax regression is most widely used classifier. Zhang et al. [13] have used stacked autoencoders for image feature extraction and softmax regression for classification. In the same area of image classification, Gao et al. [14] and Dong et al. [15] have used convolutional neural network-based feature extraction from images and classification by softmax regression. The softmax regression classifier requires very small training time as compared to widely used ANN-based classifier with the same level of accuracy.

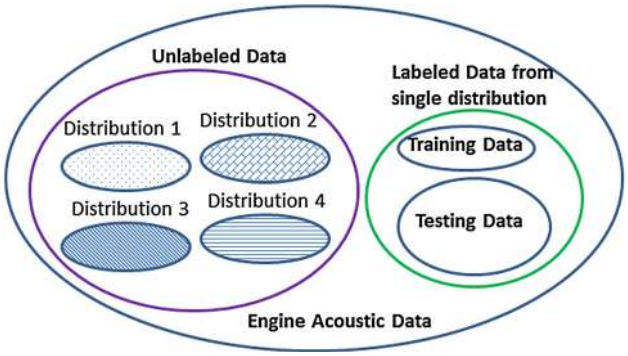

Fig. 1 Organization of data

The proposed technique was tested on the industrial IC engine data sets, with three different fault classes and one healthy class. The acoustic data were recorded at four different positions of the engine and data from each position are used independently to compute the performance of the technique. A majority voting-based criteria among all four positions are used to finally declare the type of fault in the engine.

Proposed technique

The frequency spectrum representation is the best-suited approach to represent the large size of time-domain data in a small size vector, so all acoustic signal data are transformed into the frequency spectrum by FFT. The labeled data set is divided into the training data set and testing data set. The data sets for the proposed technique are shown in Fig. 1. These are the preconditions of forming the unlabeled and labeled data sets:

1. The generating source of both the labeled and unlabeled data sets shall be the same type (or the same type of engines).
2. The unlabeled data set can be from any data distributions, but the labeled data shall be from the same data distributions.
3. The unlabeled data set can have data for any fault type.

The position of a sensor on the engine represents a distribution.

The flow diagram of the proposed technique is shown in Fig. 2. A restricted Boltzmann machine RBM1 is first trained with this unlabeled data set. The extracted features or weight matrix W_1 of this RBM1 are used as the initial weights of the RBM2.

The features or weights W_2 of RBM2 are then used to linearly transform both the testing and training data sets before being used by classifier. The principle of restricted Boltzmann machine and softmax regression-based classifier is explained in sections (A) and (C).

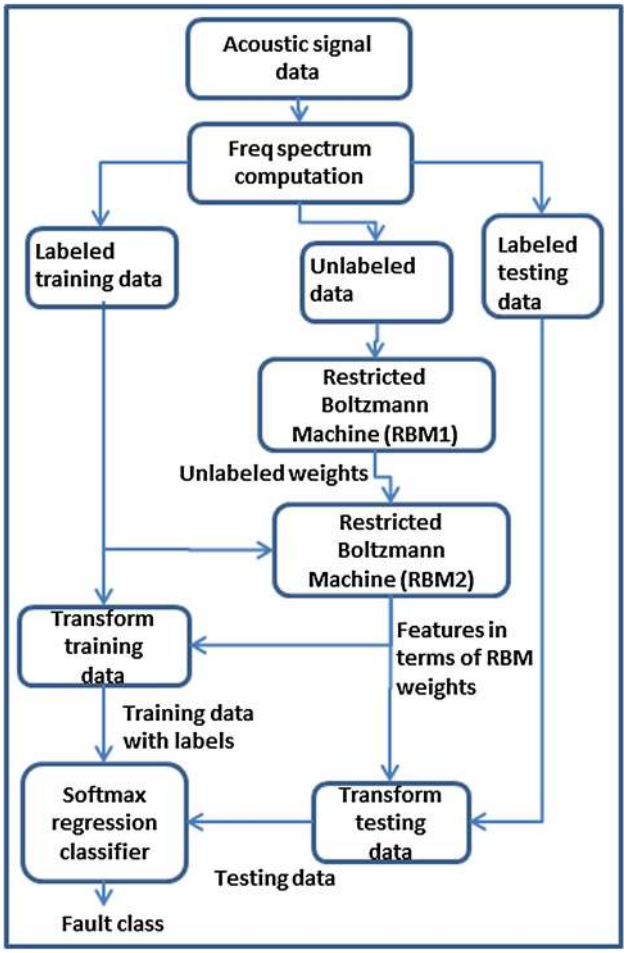

Fig. 2 Flow diagram of fault detection and classification by RBM and softmax regression classifier

Based on the above discussion, the following three types of data sets are used in the proposed technique:

1. Unlabeled data set $x_{\text{ul}}^{(i)} \in R^m$ with u numbers of data vectors.
2. Labeled training data set $x_l^{(i)} \in R^m$ with v numbers of data vectors. $\{(x_l^{(1)}, y_l^{(1)}), (x_l^{(2)}, y_l^{(2)}), \ldots (x_l^{(v)}, y_l^{(v)})\}$, where $y_l^{(i)} \in (1, 2 \ldots C)$ is the class label of each training data vector and C is number of fault classes or labels.
3. Testing data set $x_t^{(i)} \in R^m$.

Principle of restricted Boltzmann machine for unsupervised feature extraction/learning

Restricted Boltzmann machine is a stochastic neural network with a visible and hidden layer. Each unit of the visible layer is having a undirected connection with each unit of the hidden layer, with weights associated with them. Each unit of the visible and hidden layer is also connected with their respective bias units. The structure of RBM is shown in Fig. 3. The

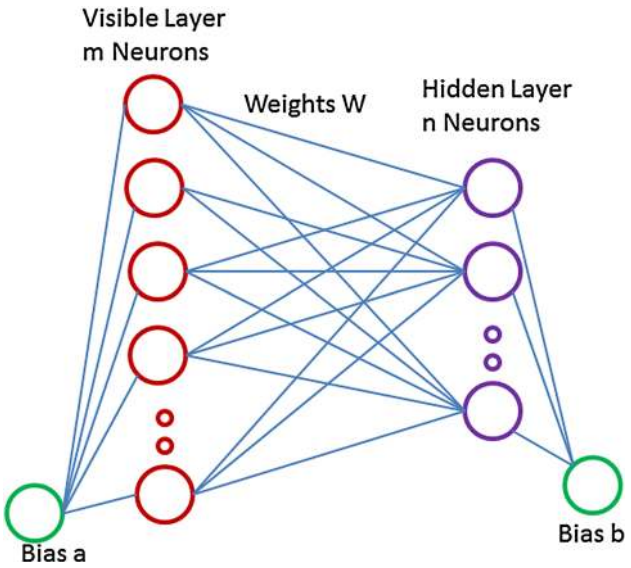

Fig. 3 Structure of RBM

RBMs do not have connections among the visible units and similarly in hidden units also. This restriction on connection makes it restricted Boltzmann machine. The state of a neuron unit in a hidden layer is stochastically updated based on the state of the visible layer and vice versa for the visible unit.

The energy function of the RBM model for visible and hidden units can be computed as

$$E(v, h) = -a^T v - b^T h - h^T W v, \tag{1}$$

where a and b are bias of the visible units and hidden units, respectively. The parameter W is weights of the connection between visible and hidden layer units. The joint probability distribution of visible units v and hidden units h of the RBM is defined by

$$P(v, h) = \frac{1}{Z} e^{E(v,h)}, \tag{2}$$

where Z is partition function and defined as sum of energy functions of over all possible configurations:

$$Z = \sum_{v',h'} e^{E(v',h')}. \tag{3}$$

The conditional probability of activation of hidden layer given the visible state v is computed as

$$P(h = 1|v) = \sigma(W^T v + b), \tag{4}$$

where function σ is logistic function. Similarly, the conditional probability of activation of visible layer given the hidden state h is computed as

$$P(v = 1|h) = \sigma(W^T h + a). \tag{5}$$

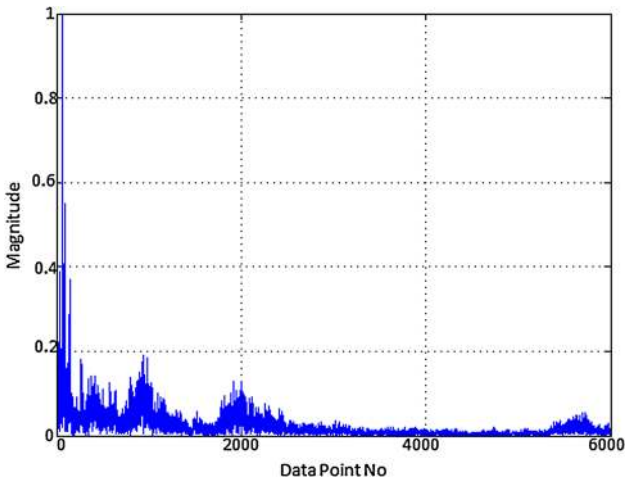

Fig. 4 Original pattern of data

On training, the RBM updates its weights W and bias a and b, to maximize the probabilities assigned to training set $x_l^{(i)}$. For training of the RBM, the contrastive divergence (CD) training method [17,18] is used. The contrastive divergence training is performed with the stochastic steepest ascent. The change of the parameter W by the CD training is given by

$$\Delta W_{ij} = \epsilon(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}}). \tag{6}$$

where parameter ϵ is learning rate and v_i is state of visible layer unit given by Eq. (5) and h_j is state of hidden layer unit given by Eq. (4). The weight matrix W is initialized by some random values and then updated by the value ΔW for each training data set. Similarly, the increments in bias are computed and the bias vectors a and b are updated. The term $\langle v_i h_j \rangle$ represents the average of the state values products. The subscript "data" is for the value of hidden state computed by Eq. (4), and subscript "recon" is for the value of visible state computed by Eq. (5). The more detail of RBM implementation and training with CD is given in the guide by Hilton [19].

The number of neurons in the visible layers is always equal to input training vector of size m, but the number of neurons in hidden layer n is selected based on the factor by which dimension of training data needs to be reduced. The training data matrix of size $m \times v$ is reduced to feature matrix W of size $m \times n$, where $n \ll v$. The weight matrix W has n linearly independent basis vectors and each represents a unique feature learned from the data.

In a typical case of RBM2 with 50 hidden neurons, there are 50 feature vectors in the features matrix W. A typical pattern of fault data is shown in Fig. 4 and plots of some of the typical learned feature patterns from the features matrix W are shown in Fig. 5.

In the proposed technique, the structure of both RBM1 and RBM2 is exactly same. Initially, the RBM1 is trained with

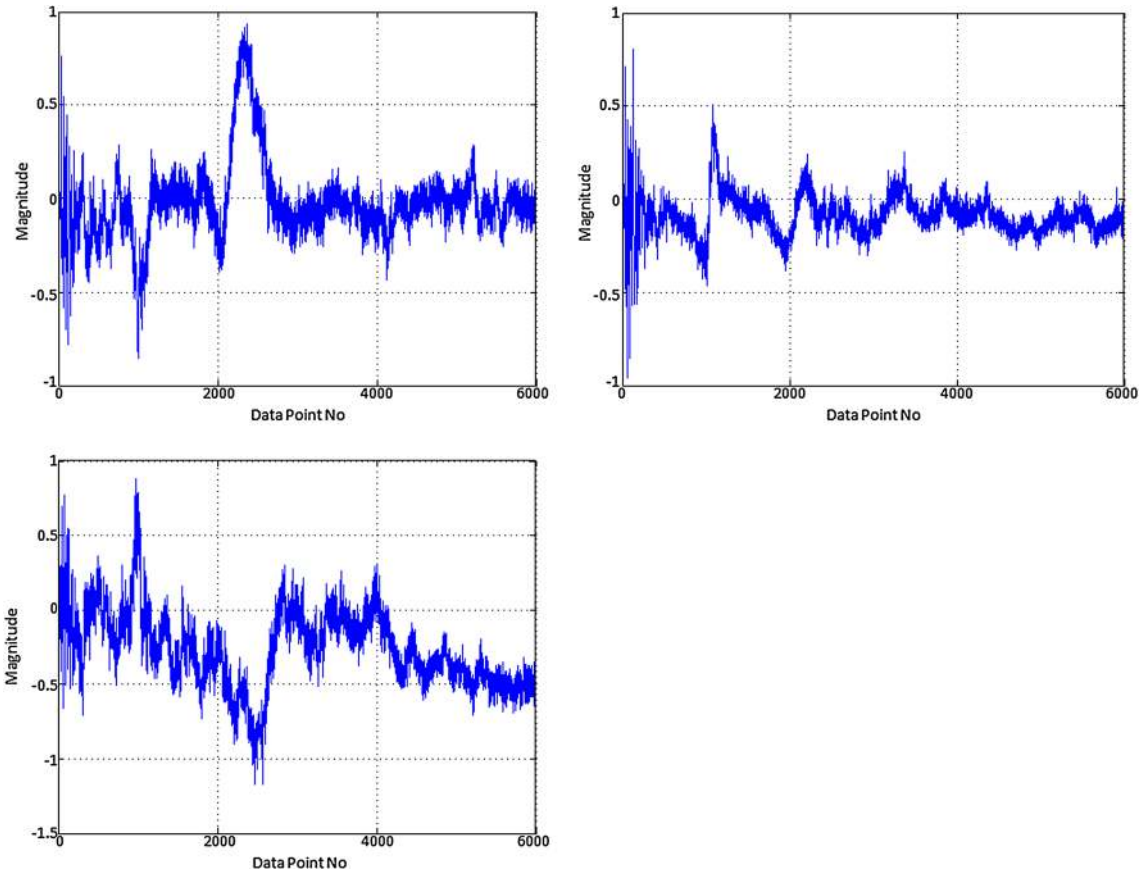

Fig. 5 Typical learned feature patterns

the unlabeled data set $x_{\text{ul}}^{(i)}$ and the learned weight matrix W_1 and bias vectors a_1, b_1 are used as initial values of the W_2, a_2, and b_2 of the RBM2. RBM2 is then trained with the training data set $x_l^{(i)}$ and the learned weight matrix W_2 is further used for transformation. The learning rate, $\epsilon = 0.01$, was used in training of both RBMs.

Transformation of training and testing data by extracted features

The feature matrix W_2 is used to linearly transform the input training and testing data vectors into lower dimensional feature vectors. The training data vector $x_l^{(i)} \in R^m$ and testing data vector $x_t^{(i)} \in R^m$ are transformed into $\hat{x_l}^{(i)} \in R^n$ and $\hat{x_t}^{(i)} \in R^n$, respectively, as follows:

$$\hat{x_l} = {W_2}^T x_l, \tag{7}$$

$$\hat{x_t} = {W_2}^T x_t. \tag{8}$$

These transformed vectors $\hat{x_l}^{(i)}$ and $\hat{x_t}^{(i)}$ are now represented as the weighted linear combination of the feature vectors from W_2. In other words, the features of $x_l^{(i)}$ and $x_t^{(i)}$ are compressed and represented in terms of these learned features. The new training data set $\hat{x_l}^{(i)}$ with v number of labeled training data vectors is used to train the softmax regression classifier.

The size of the transformed training and testing data vectors is n, which is very less than original size m. This size reduction is due to the number of hidden layer neurons that are less than the number of input layer neurons or $n \ll m$. This way the proposed technique improves the classification performance by enhancing the feature representation and reducing the size of the training and testing data vectors. In typical case, an input training and testing spectrum vector of size 6000 is reduced to hidden layer size of 50 after transformation. The small size of training vector requires small set of weight in a classifier and the cost function is easy to optimize for these small set of weights. This improves the classification performance with reducing training time.

Principle of the softmax regression classifier

The softmax regression is a generalization of the logistic regression [9,11], where the output class labels are multi-

class $y_i \in (1, 2, \ldots k)$, instead of binary output classes. The input training set for softmax regression with v numbers of data vectors $\{(x_1, y_1), (x_2, y_2), \ldots (x_v, y_v)\}$, where $x_i \in R^n$. In the softmax regression-based classifier, the probability $P(Y = j/X)$ of X belonging to each class from set of k classes is given as

$$P\left(y_i = j|x_i; \theta\right) = \frac{e^{\theta_j^T x_i}}{\sum_{l=1}^{k} e^{\theta_l^T x_i}}, \tag{9}$$

where the parameters $j = 1, \ldots, k$ and $Y = [y_1, y_2, \ldots, y_k]$ are output class. The input variables to this probability function are feature vector $X = [x_1, x_2, \ldots, x_v]$, and the weight or model parameter $\theta = [\theta_0, \theta_1, \ldots, \theta_k \in R^n]$ of softmax regression model. The generalized softmax regression cost function is defined as

$$J(\theta) = -\frac{1}{v}\left[\sum_{i=1}^{v}\sum_{j=0}^{1} 1(y_i = j) \log P(y_i = j|x_i; \theta)\right]. \tag{10}$$

This softmax regression cost function has no closed form way to minimize the cost value, so the iterative algorithm, gradient descent is used. To make the softmax Regression cost function strictly convex, so that it can converge to a global minimum, a weight decay term is added. The modified cost function with its gradient is given as follows:

$$J(\theta) = -\frac{1}{v}\left[\sum_{i=1}^{v}\sum_{j=0}^{k} 1(y_i = j) \log P(y_i = j|x_i; \theta)\right] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2, \tag{11}$$

$$\nabla_{\theta_j} J\left(\theta\right) = -\frac{1}{v}\sum_{i=1}^{v}\left[x_i\left(1\left\{y_i = j\right\} - p\left(y_i = j|x_i; \theta\right)\right)\right] + \lambda\theta_j, \tag{12}$$

where the weight decay parameter λ shall always be positive. The weight parameters are updated by $\theta_j = \theta_j - \alpha\nabla_{\theta_j} J\left(\theta\right)$ for $j = 1, \ldots, k$. The weights θ of softmax regression are initialized with random values, and these weights are updated with each training vector $\hat{x_l}^{(i)}$, to minimize the value of the cost function. The number of weight vectors $[\theta_0, \theta_1, \ldots, \theta_k]$ in the softmax regression is equal to the number of output classes and size of each weight vector $\theta_k \in R^n$ is equal to the size of input data vector. In this case, the size of input data vector is equal to the number of hidden layer neurons in RBM. In the implementation of softmax regression, $\lambda = 0.001$ was used.

Fig. 6 Experimental setup of IC engine

Experimental setup and results

The proposed technique was tested on data sets recorded in the industrial environment from single cylinder IC engines of a commercial two wheeler manufacturing company. In the test rig, four PCB 130D20 piezoelectric microphones were placed at four different parts or assemblies of the engine to record the audio signatures, as shown in Fig. 6. The speed of rotation of the engine was kept at 40 Hz with the accuracy of $\pm 2\%$.

Organization of data and testing procedure

For both the labeled and unlabeled data sets, fault signal was recorded from all four positions of the sensor. For the labeled data set, the acoustic fault data are recorded for three different types of seeded faults and one normal operation, as shown in Table 1. For the unlabeled data set, three different types of faults are seeded randomly. These seeded faults for the unlabeled data set are different from the faults seeded for the labeled data set. Each position of the sensor on the setup represents a data distribution D_i. The unlabeled data from distribution D_1 to D_4 are merged to form the unlabeled data set x_{ul}. This common unlabeled data set x_{ul} is used by RBM1 to compute unlabeled weights W_1 and bias a_1 and b_1. These unlabeled weights and bias are then used as the initial weight and bias of the RBM2. This technique is tested separately for

Table 1 Types of fault Seeded and number of data sets for each fault type

S. no	Fault type	Number of data sets	Category of data set
1	PGW (Primary gear whining)	64	Labeled
2	MRN (Magneto rotor noise)	65	Labeled
3	TAPPET	59	Labeled
4	Healthy engine	60	Labeled
5	CCN (Cam chain noise)	60	Unlabeled
6	CHN (Cylinder head noise)	40	Unlabeled
7	PGD (Primary gear damage)	57	Unlabeled

each sensor position, with their respective labeled data sets and with this common unlabeled data set.

Table 1 shows the seeded faults and number of data sets recorded for each fault type. The details of each fault are described in [4,20,21]. There are total 248 labeled data sets that are recorded for each sensor position. In testing of this technique, the last three faults from Table 1 were used as part of the unlabeled data set and the rest were part of labeled data set. For testing, the labeled data set for each position of the sensor is divided into different ratios of training and testing data set, as shown in Table 2.

In this work, the majority voting (MV) is the majority of classification type among all four sensor positions. If the classification type has more than two votes for a class, then the classification belongs to that particular class. In addition, if there is a tie between votes, then the classification is assumed from incorrect class only. The classification performance is depicted in %, total correct classification * 100/total test cases, in all the tables.

Table 2 Classification performance with labeled data in % with different training and testing data set division ratios

Ratio (%)	Pos 1	Pos 2	Pos 3	Pos 4	MV	MV SD
5–95	84.5	79.15	72.23	86.99	81.50	7.80
15–85	98.65	96.8	91.59	98.77	99.35	0.75
25–75	99.64	99.22	95.37	99.51	99.85	0.47
35–65	99.93	99.62	97.08	99.79	99.99	0.09
50–50	99.99	99.93	98.61	99.72	99.99	0.08
75–25	100	99.98	99.36	99.49	100	0

Table 3 Positionwise classification performance of each fault in %

Fault type	Pos 1	Pos 2	Pos 3	Pos 4	Majority voting
PGW	96.30	92.59	92.59	100	100
MRN	100	100	100	98.18	100
TAPPET	100	96	70	96	98
Healthy engine	100	94.12	96.08	100	100

Results for different training and testing data division ratios

To test this technique, the labeled data set is divided into different ratios of training and testing data sets, as shown in Table 2. In each division ratio, the training and testing data are randomly selected and classification performance is computed for 100 iterations. Table 2 shows the average classification performance of these 100 iterations. The classification performance of each position with different division ratios of the labeled data set in training and testing data is shown in Table 2, along with majority voting (MV) among all four positions with its standard deviation (SD). The results in Table 2 are with hidden layer size of 50 neurons in both RBM1 and RBM2.

In typical ratio of (15–85%), where only 38 training data sets for all four faults types were used for training of the RBM2. The classification performance is more than 90% for each position, as shown in Table 2. The performance after majority voting, among all positions, is 99.35%. With the increase in the size of the training data, there is a small improvement in the MV classification performance and its SD. The small value of SD shows the consistency in MV classification accuracy or small variations in the MV classification accuracy.

Table 3 shows a typical case of 15–85% division ratio (38 training sets and 210 testing sets). The individual classification performance for each fault type is more than 90%, except one case TAPPET fault in position 3. In this case, the overall MV classification performance is 209 correct classifications out of 210 test cases. In all 210 test cases, only one case was wrongly classified by two classifiers on majority voting.

From the above analysis, it can be concluded that the proposed technique works very well in the industrial environment, with performance more than 99%. This performance was achieved with the 40 Hz rotation speed of the engine with variations in the range of ±2%. The performance in case of 5–95% is only 81.50%; this is due to insufficient training data sets for the training of the RBM2 and softmax regression classifier. In this case, total numbers of training data sets are only 12 for all four fault classes. This amount is too small to

Table 4 Classification performance without unlabeled data for different division ratios

Ratio (%)	Pos 1	Pos 2	Pos 3	Pos 4	MV	MV SD
5–95	62.43	61.52	55.64	71.78	52.56	9.87
15–85	89.04	86.12	85.27	92.93	91.75	4.01
25–75	94.10	93.37	91.07	96.34	96.82	2.04
35–65	96.16	96.01	94.04	98.27	98.85	1.13
50–50	97.53	97.41	95.94	98.89	99.55	0.70
75–25	98.05	99.20	97.25	99.20	99.84	0.49

train any machine learning algorithm. As compared to unsupervised feature extraction based on the sparse autoencoder proposed in [10], the proposed technique reduces the requirement of training data significantly with large improvement in the performance. Due to use of the unlabeled features for as initial weights of RBM2, the cost function of the RBM2 does not get trapped in local minima and reaches global minima very fast with small number of training examples.

Performance comparison with and without unlabeled data

The classification performance without unlabeled data (WoUD) is shown in Table 4. By comparing the results from Table 4 to results with unlabeled data (WiUD) from Table 2, the performance of WiUD is superior for the same size of training data sets. There is too much variation in MV accuracy in case of WoUD as compared to the result of WiUD. The initialization of the initial weights of RBM2 by the unlabeled features improves the learning of the RBM2 by achieving global minima with small set of training data, and this reduces the inconsistency in the results. The use of unlabeled data reduces the requirement of labeled training data and provides high performance with consistency in results. The unlabeled data also form the same source types with different types of distribution, but poses the similar types of the relations or features as in the training and testing data. Therefore, RBM1 also extracts somewhat similar types of the features as available in training and testing data. In addition, initialization of RBM2 weights with this makes minimization of cost function fast and consistent. The comparisons of the performance with and without unlabeled weights are shown in Figs. 7, 8, 9, and 10 for each position. Figure 11 shows the majority voting performance with and without unlabeled weights. It can be seen from these figures that there is a significant improvement in performance with use of unlabeled weights. With small number of training data sets, the use of unlabeled weight improves performance significantly with consistency in results. The initialization of RBM2 initial weights with

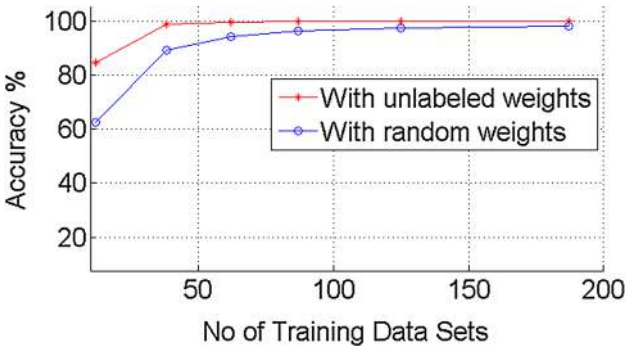

Fig. 7 Performance with number of training examples for Position 1

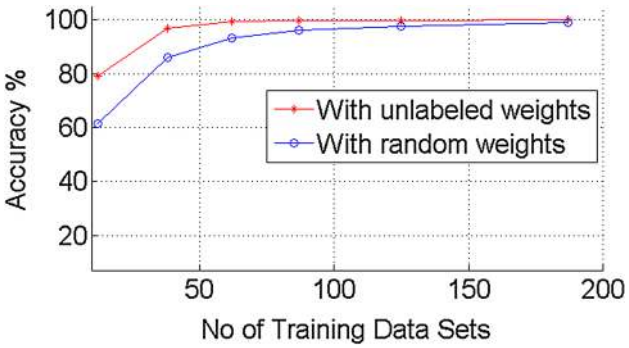

Fig. 8 Performance with number of training examples for Position 2

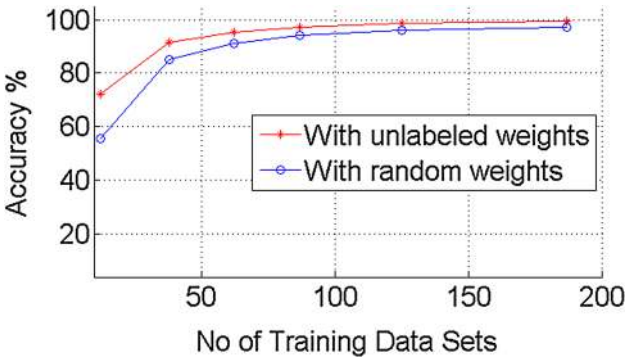

Fig. 9 Performance with number of training examples for Position 3

Classification performance and complexity with different sizes of hidden layer in RBM1 and RBM2

Table 5 shows that increasing the number of hidden layer neurons in case of WoUD increases the classification accuracy for the given size of training data with increases in the computation time. There is an improvement in the SD of MV also. In the case of WiUD for the same size of training data, the increase in the size of hidden layer improves the consistency in classification accuracy but at the cost of increased computation time or time complexity. Increasing the number of neurons in the RBM increases the number of feature extracted from data, and this improves the classification per-

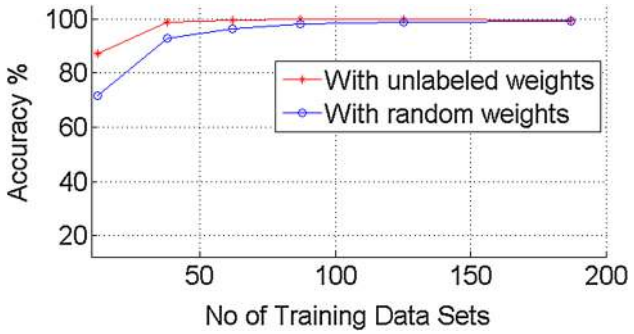

Fig. 10 Performance with number of training examples for Position 4

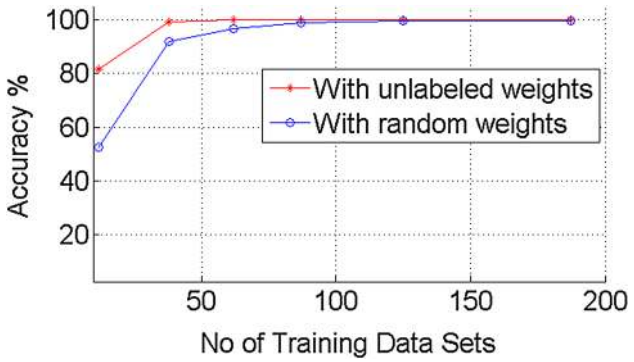

Fig. 11 Majority voting performance with number of training examples

formance a bit, but makes the cost function minimization more difficult. Because more the number of neurons, more the number of weights and this makes minimization of the cost function more difficult and requires more time to achieve the global minima.

Therefore, the hidden size 50 neurons provide an optimal choice for accuracy, time, and SD for WiUD. Table 5 shows that for a given size of training data, the use of unlabeled data provides more accuracy and small SD in accuracy. The accuracy with consistency is more important in fault classification than the extra computation time due to the unlabeled data. Once the RBM2 and classifier have learned the weights, then to classify a new test data set does not takes any time. The new test data set first converted into spectrum vector and then transformed by the RBM2 weights before used by the classifier.

Comparison with PCA-based feature extraction

For comparison of the proposed RBM-based feature extraction with PCA, the 50 most significant eigen vectors of the training data set after PCA were used as the features training and testing data. From Table 6, it can be seen that the proposed RBM-based feature extraction has outperformed the PCA-based feature extraction with consistency in results for the same size of the training data set size. At the small size of training data set (15–85%), the performance of the proposed technique is far better than PCA. PCA is a linear operation and does not extracts the complex features in the data, but the RBM has a nonlinear sigmoid function in its basic unit neuron. This helps the RBM to learn complex nonlinear relations in the data more efficiently with small size of the training data. This makes the classification more consistent with small size of training data. As the number of training data increases, the consistency in the results by PCA improves but still not at the level of RBM (Table 6).

Comparison with existing techniques

Yadav et al. [20] have proposed an FFT and correlation-based technique using acoustic data from the same type of IC Engine Test Rig. In this work, the final classification accuracy for four different types of fault classes was less than 93%. The classification accuracy for CHN fault was 80%, and for MRN fault, it was 93%. In this technique, the faulty engine was compared with a prototype engine, so no classifier was used. Nidadavolu et al. [21] also proposed a fault detection technique based on empirical mode decomposition (EMD) and Morlet wavelet for the same type of IC engine. The overall classification accuracy for the proposed technique was less than 90% for each fault type and each position. In this work, they have used five different types of fault classes with total 540 data sets. Out of these 540 data sets, 70% were used for training of an artificial neural network (ANN) classifier and rest 30% were used for testing. In a similar type of fault detection by Wu et al. [3], the feature extraction was done using WPT and 'Shannon entropy' from acoustic data of the GDI (gasoline direct-injection) engine. They have recorded 150 experimental data for each operating condition of the engine for six different types of faults. Out of these 150 data sets, 30 were used for training and the remaining were used for testing. The average classification accuracy for an ANN classifier for the different operating conditions of the engine was around 95%. Yadav et al. [4] has proposed a spectrogram based statical feature extraction technique for the same type of test rig. The majority voting accuracy of their technique was less than 93% for all fault classes. In this work, they have used 400 training data sets to train ANN classifier with seven different types of fault classes and 200 data sets for testing. For the same data set, Chopra et al. [10] have proposed a unsupervised feature extraction technique by a sparse autoencoder. This technique has 98% classification accuracy for four different types of fault classes without any unlabeled data for 62 training data sets. The proposed techniques overcome the problem of random initialization [10] of the sparse-autoencoder weights with more accuracy.

As compared to the above-discussed techniques, the proposed technique is at par with the other techniques in terms

Table 5 Classification and timing performance (in s) with SD of MV for different sizes of hidden layer for typical division ratio of 25–75%

Hidden size	Accuracy WoUD	Accuracy WiUD	Time WoUD	Time WiUD	SD WoUD	SD WiUD
25	89.51	99.38	9.11	42.45	5.77	1.01
50	93.57	99.78	16.85	64.51	5.05	0.57
75	96.84	99.78	25.91	88.83	4.11	0.57
100	98.47	99.81	32.69	105.85	3.25	0.44
125	99.83	99.84	43.56	135.79	0.36	0.29
150	99.72	99.92	49.50	152.05	0.48	0.19
175	99.83	99.94	58.67	176.36	0.33	0.22
200	99.96	99.86	59.20	181.11	0.15	0.28

Table 6 Comparison table for MV classification performance and SD for different division ratios with PCA

Division ratio (%)	Accuracy WiUD	Accuracy PCA	SD WiUD	SD PCA
5–95	81.50	72.95	7.80	6.47
15–85	99.35	95.74	0.75	2.43
25–75	99.85	99.19	0.47	0.80
35–65	99.99	99.64	0.09	0.48
50–50	99.99	99.92	0.08	0.23
75–25	100	99.92	0	0.34

of classification performance without any hand-engineered feature extraction. This technique requires less training data as compared to other techniques available in the literature.

In the industrial environment, where a lot of noise is there in recordings of sensor data, the RBM-based feature extraction is very much successful. This way the proposed technique proves its robustness for the industrial environment. The use of the unlabeled data reduces the requirement of labeled training data significantly along with significant enhancement in the performance and consistency in the results.

The implementation and analysis of this technique were done on Matlab-2013, on an Intel i5 CPU with 8GB RAM.

Conclusion

The proposed restricted Boltzmann machine and softmax regression classifier-based fault detection and classification technique were tested on the industrial IC engine data sets. This technique performs very well on industrial acoustic data of IC engines. The major advantage of this technique is that it does not require any hand-engineered feature extraction from acoustic data and still provides a very good performance with the small set of labeled training data. The performance of the technique for four different fault classes is more than 99%. In a typical example with 38 training data set and 210 testing data set, this technique is able to classify 209 test data sets correctly.

References

1. Bloch HP, Geitner FK (1997) Machinery failure analysis and trouble shooting. Gulf, Houston
2. Yen GY, Lin KC (1999) Wavelet packet feature extraction for vibration monitoring, neural networks. IJCNN 5:3365–3370
3. Wu JD, Liu CH (2009) An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network. Expert Syst Appl 36(3):4278–4286 (Part 1)
4. Yadav SK, Kalra PK (2010) Automatic fault diagnosis of internal combustion engine based on spectrogram and artificial neural network. In: Proceedings of the 10th WSEAS international conference on robotics, control and manufacturing technology. Hangzhou, pp 101–107, 11–13 April 2010
5. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24 th international conference on machine learning, Corvallis
6. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507
7. Taylor GW, Hinton GE, Roweis S (2007) Modeling human motion using binary latent variables. In: Proceeding of advances in neural information processing systems 19 (NIPS 2006). MIT Press, Cambridge
8. Teh YW, Hinton GE (2000) Rate-coded restricted Boltzmann machines for face recognition. In: Proceeding of advances in neural information processing systems 13 (NIPS 2000). MIT Press, Cambridge
9. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York, pp 205–213
10. Chopra P, Yadav SK (2015) Fault detection and classification by unsupervised feature extraction and dimensionality reduction. Complex Intell Syst 1:25–33
11. Bhning D (1992) Multinomial logistic regression algorithm. Ann. Inst Stat Math 44(1):197–200

12. Krishnapuram B, Carin L, Figueiredo MAT, Hartemink AJ (2005) Sparse multinomial logistic regression: fast algorithms and generalization bounds. IEEE Trans Pattern Anal Mach Intell 27(6):957–968
13. Zhang H, Zhu Q (2014) Gender classification in face images based on stacked-autoencoders method. In: 7th international congress on image and signal processing, IEEE
14. Gao J, Yang J, Zhang J, Li M (2015) Natural scene recognition based on convolutional neural networks and deep Boltzmannn machines. IEEE international conference on mechatronics and automation (ICMA), IEEE
15. Dong Z, Pei M, He Y, Liu T, Dong Y, Jia Y (2014) Vehicle type classification using unsupervised convolutional neural network. 22nd international conference on pattern recognition, IEEE
16. Raina R, Battle A, Lee H, Packer B, Ng AY (2007) Self-taught learning: transfer learning from unlabeled data. In: Proceedings of the 24th International Conference on Machine Learning (ICML). Corvalis, pp 759–766
17. Carreira-Perpinan M, Hinton G (2005) On contrastive divergence learning. In: 10th international workshop on artificial intelligence and statistics (AISTATS-2005, pp 59–66)
18. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
19. Hinton GE (2012) Neural networks: tricks of the trade. Lect Notes Comput Sci 7700:599–619
20. Yadav SK, Tyagi K, Shah B, Kalra PK (2011) Audio signature-based condition monitoring of internal combustion engine using FFT and correlation approach. IEEE Trans Instrum Meas 60(4):1217–1226
21. Nidadavolu SVPS, Yadav SK, Kalra PK (2009) Condition monitoring of internal combustion engines using empirical mode decomposition and Morlet wavelet. In: Proceedings of 6th international symposium on image and signal processing and analysis (ISPA 2009). IEEE, pp 65–72