Mechanical Systems and Signal Processing 117 (2019) 170–187

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems

Te Han [a,b], Chao Liu [a,c,*], Linjiang Wu [d], Soumik Sarkar [d], Dongxiang Jiang [a,b]

[a] Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China
[b] State Key Laboratory of Control and Simulation of Power System and Generation Equipment, Tsinghua University, Beijing 100084, China
[c] Key Laboratory for Thermal Science and Power Engineering of Ministry of Education, Tsinghua University, Beijing 100084, China
[d] Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA

ARTICLE INFO

Article history:
Received 28 February 2018
Received in revised form 24 May 2018
Accepted 24 July 2018
Available online 4 August 2018

Keywords:
Spatiotemporal pattern network
Convolutional Neural Network (CNN)
intelligent diagnosis

ABSTRACT

The machine fault diagnosis is being considered in a larger-scale complex system with numerous measurements from diverse subsystems or components, where the collected data is with disparate characteristics and needs more prevailing methods for data preprocessing, feature extraction and selection. This work presents a novel diagnosis framework that combines the spatiotemporal pattern network (STPN) approach with convolutional neural networks (CNN) to build a hybrid ST-CNN scheme. The proposed framework is tested on two data sets for diagnosing unseen operating conditions and fault severities respectively, to evaluate its generalization ability, which is essential for the application in machine fault diagnosis as not all of the aforementioned scenarios have sufficient labeled data to train a model. The results show that the proposed ST-CNN framework outperforms or is comparable to shallow methods (support vector machine and random forest) and 1D CNN. Through visualizing the activations, it is verified that the spatial features can elevate the diagnosis accuracy, and more general features are determined by the proposed approach to form an adaptive classifier for diverse operating conditions and different fault severities.

1. Introduction

During the evolution of the operation and maintenance strategy of machinery, the condition classification, which tries to distinguish the anomaly from normal and identify the types of anomalies, is an essential task in condition monitoring, diagnosis, prognostics, and health management. With the development of sensing and data availability, the information that can be employed for the condition classification is greatly scaled up in terms of the types of measurements and the volume of data. Numerous examples of this include bearing, gearbox, and rotor system. Besides the widely used vibration (e.g., displacement, velocity, and acceleration), acoustic emission [1–3], sound [4], voltage and current [5–9], and temperature [10,11] are more and more applied in condition classification for diagnosis and prognostics [12,13]. The diverse types of measurements describe the system in different attitudes, while they are probably with different characteristics comparing to the vibration measurements in terms of data processing, feature extraction and selection approaches [14]. Also, the condition classification is being considered in an increasing extent of system, where more measurements are collected from different

* Corresponding author.
E-mail address: cliu5@tsinghua.edu.cn (C. Liu).

https://doi.org/10.1016/j.ymssp.2018.07.048
0888-3270/

subsystems or components. In this scenario, the condition classification performance is expected to be improved with more information sources, especially when the dependency of the subsystems (the relationship of the fault location to other locations) is taken into account. The fault occurred in wind turbine is a case with this kind of dependency, where the faults attributed to wind wheel (e.g., aero-asymmetry and icing blade) are difficult to be diagnosed by the vibration measurements from the drivetrain (e.g., accelerator on bearing or gearbox). However, when wind speed, rotor speed, and generated power output are provided, the aforementioned faults can be more easily identified [15,16]. In this context, a robust condition classification framework is crucial for the complex systems, which can (i) deal with disparate measurements and different types of data, and (ii) capture the dependency between the components of the system and fuse the features of individual subsystems. Furthermore, as the equipment operates on diverse settings and the combinations of the operating conditions in different subsystems are huge. As a result, it is highly appreciable that the diagnosis algorithm can handle different operating conditions, especially when the model is trained on one operating condition and applied in another.

Intensive research has been conducted on the condition classification of machinery with acceleration widely used. Data preprocessing, feature extraction, and feature selection approaches are proposed for improving the accuracy [17–22]. These feature extraction methods are mostly focused on the vibration (acceleration), and are proven to be extremely useful for the faults that occurred in the bearing, gearbox, etc. However, such kind of feature extraction is based on the domain knowledge of the specific device, namely handcrafted features [23], which may not be applicable in other types of measurements (e.g., acoustic emission, temperature). In this context, an adaptive feature extraction approach is required that can deal with diverse types of measurements.

With the purpose of avoiding manual feature extraction, deep learning approaches have been successfully applied for fault diagnosis, where raw data (usually time-series) is inputted to the deep structures for condition classification [24–28]. Among them, Convolutional Neural Network (CNN) is shown to be highly reliable in dealing with time-series data including univariate [26] and multivariate ones [27,29]. However, the computational cost for this type of model is high because the input data is usually high dimensional, especially when the input is multivariate. As a result, compressed sensing methods are applied to reduce the dimensions of the input [25]. Although the compressed sensing approaches are effective in dimensional reduction, they also sacrifice the temporal features in the original space which may lose the essential features of the faults with pulse impacts (e.g., gear tooth fault in the gearbox).

The traditional feature extraction methods and deep learning structures are mainly focused on the extraction of temporal features (within each individual time-series), the spatial features (that represent the dependency between multiple measurements) are rarely discussed and adopted for the diagnosis. The deep learning structure in the multivariate scenario intends to learn the joint distribution of multiple time-series data. However, the spatial features are still difficult to be learnt as of the characteristics of local receptors (in CNNs) and dimension skewing between the temporal and spatial resolutions of the inputs. As discussed above, the spatial features are important to diagnose the conditions in which the dependency is an indicator of system status, such as wind speed and wind power for a wind turbine.

Motivated by forming a more adaptive feature learning approach that can extract both spatial and temporal features from diverse types of data in an efficient manner, this work presents a spatiotemporal feature learning framework, built on spatiotemporal pattern network (STPN) [30,31], to process multiple time-series data in complex systems and learn spatiotemporal features. The learnt features are then connected to a deep learning structure (CNNs in this work) to implement the condition classification.

The contributions of this work include: (i) proposing an adaptive spatiotemporal feature learning approach for extracting both spatial and temporal features from diverse types of time-series data, (ii) data abstraction process applied in STPN avoids the manual feature extraction and improves the adaptivity of the proposed approach, (iii) the proposed framework comprises STPN and CNNs (ST-CNN) that shows to be more applicable in unseen operating conditions and fault severities, (iv) the presented framework outperforms 1D CNN and shallow methods (Support Vector Machine–SVM and Random Forest–RF) and is computational efficient comparing with 1D CNN, and (v) the interpretation of the learnt features by ST-CNN is explored via Gradient-weighted Class Activation Mapping (Grad-CAM) and the spatial features are shown to be able to elevate the diagnosis accuracy.

The remaining sections are organized as follows. Section 2 provides background and preliminaries including the definition of STPN and CNN basics. The proposed diagnosis framework with STPN and CNN is presented in Section 3. Case studies on two data sets are illustrated in Section 4 as well the discussions and future work. Finally, the paper is summarized and concluded in Section 5.

2. Background and preliminaries

2.1. Spatiotemporal Pattern Network (STPN)

Symbolic dynamics filtering (SDF) is formed with the assumption that a symbol sequence (generated by the time-series) can be modeled as a Markov chain of order D (namely the depth of the Markov machine in the presented method) which captures the characteristics of the time-series [32,33]. Data abstraction is first implemented in SDF, to generate the symbol sequences of the time-series (sub-systems) including data preprocessing and discretization. The discretization transforms the time-series into a symbol sequence (namely partitioning in the symbolic dynamics literature). For a system consisting

of multivariate time-series, let $\mathbb{X}$ denote a set of partitioning functions, $\mathbb{X}: X(t) \rightarrow S$, which convert a general dynamic system (time series $X(t)$) into a symbol sequence S using an alphabet set Σ [34]. Depending on different objective functions, several partitioning approaches have been proposed in the literature, such as uniform partitioning (UP), maximum entropy partitioning (MEP), maximally bijective discretization (MBD) and statistically similar discretization (SSD) [35].

Built on SDF, an STPN is defined from a multivariate perspective. Consider the multivariate time series, $X = \{X^{\mathbb{A}}(t), t \in \mathbb{N}, \mathbb{A} = 1, 2, \ldots, f\}$, where f is the dimension or number of variables of the time series. With partitioning, the symbol sequences S are generated, and then a probabilistic finite state automaton (PFSA) is defined to describe the subsequent states and transition probabilities among them via D-Markov machine and xD-Markov machine [36–38]. To illustrate the formulation of D-Markov machine and xD-Markov machine, two symbol sequences S^a and S^b are illustrated in Fig. 1, which represent two sub-systems or two time-series. Two state sequences are then formed based on the symbol sequences. When $D = 1$, the state sequences and the symbol sequences are equivalent. The state transition matrices Π^a and Π^b are generated using the D-Markov machines, which represent sub-systems a and b respectively. Similarly, cross state transition matrices Π^{ab} and Π^{ba} are defined using the xD-Markov machines, which represent the dependencies of b on a and of a on b respectively. The pairwise dependencies may not be symmetric, i.e., Π^{ab} and Π^{ba} are not necessarily the same. Here, the features from D-Markov machines are noted as the atomic patterns (APs) and the features from xD-Markov machines are noted as the relational patterns (RPs) (as shown in the bottom panel of Fig. 1).

With this setup, an STPN is defined as:

Definition. A PFSA based STPN is a 4-tuple $W_D \equiv (Q^a, \Sigma^b, \Pi^{ab}, \Lambda^{ab})$ (a, b represent nodes of the STPN) [39]:

1. $Q^a = \{q_1, q_2, \ldots, q_{|Q^a|}\}$ is the state set corresponding to symbol sequences S^a.
2. $\Sigma^b = \{\sigma s_0, \ldots, \sigma s_{|\Sigma^b|-1}\}$ is the alphabet set of symbol sequence S^b.
3. Π^{ab} is the symbol generation matrix (size $|Q^a| \times |\Sigma^b|$), the ij^{th} element of Π^{ab} denotes the probability of finding the symbol s_j in the symbol sequence S^b when a transition is made from the state q_i in the symbol sequence S^a; while self-symbol generation matrices are noted as atomic patterns (APs) i.e., when $a = b$, cross-symbol generation matrices are noted as relational patterns (RPs) i.e., when $a \neq b$.
4. Λ^{ab} denotes a metric that represents the importance of the pattern for $a \rightarrow b$ which is a function of Π^{ab}.

2.2. Convolutional Neural Network (CNN)

CNN, as a type of deep networks, has been successfully applied in image processing, video processing, and natural language processing [40–42]. CNNs intend to learn the local neighborhood matching for data dimension reduction with nonlinear mapping, where the parameters to be learnt are lowered because of sharing weights in the filters of convolutional layers [43].

For the fault diagnosis applications, 1D CNN is usually adopted which naturally fits the characteristics of the time-series data. In this scheme, the features are learnt through convolutional layers in the temporal direction and the fully connected layers are applied to further reduce the dimensionality of the features and aligning the most important features for the purpose of classification. A 2D CNN is often used for 2D data (e.g., images), where the filter is with two dimensions (while the filter of 1D CNN is with one dimension). Typically, a 2D CNN consists of several convolutional layers and fully-connected layers. A brief description of convolutional layers and fully-connected layers is provided as follows for completeness.

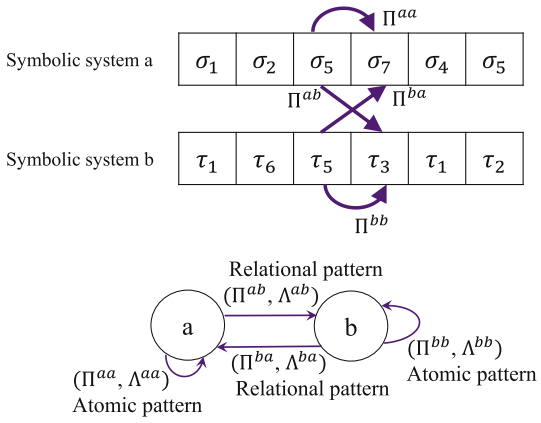


Fig. 1. Extraction of atomic patterns and relational patterns (with D-Markov machine and xD-Markov machine respectively and $D = 1$, i.e., states and symbols are equivalent) to model individual sub-system behavior and interaction behavior among different sub-systems.

2.2.1. Convolutional layer

At each convolutional layer, a number of filters are applied to convolve with input signal, and the different location on the signal will obtain a different feature activation, referred as feature map. After convolution operation, the feature maps are processed by nonlinear activation function to improve the representation capacity. This process can be described as:

$$h_n^r = \Phi(\sum_m v_m^{r-1} * K_n^r + b_n^r) \tag{1}$$

where h_n^r is the *nth* output of convolutional layer r, and n represents the filter number of layer r; v_m^{r-1} is the *mth* output of previous layer $r-1$; $*$ represents the convolution operation and K_n^r means the *nth* filter kernel of current layer r; b_n^r is the bias of current kernel; and $\Phi(\cdot)$ denotes the nonlinear activation function.

2.2.2. Fully-connected layer

The fully-connected layer is same as the layer in multi-layer neural network, where each neuron in one layer connects to every neuron in another layer. After one or several hidden layers, a classification layer, generally using softmax function, is on top of the whole network. The output of softmax function can be defined as:

$$O_j = \begin{bmatrix} P(y=1|x;\theta \\ P(y=2|x;\theta \\ \cdots \\ P(y=k|x;\theta \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} \exp(\theta^j x)} \begin{bmatrix} \exp(\theta^1 x) \\ \exp(\theta^1 x) \\ \cdots \\ \exp(\theta^k x) \end{bmatrix} \tag{2}$$

where k is the number of categories and θ^j x is the parameters of the classification layer.

3. Diagnosis framework with spatiotemporal feature learning and Convolutional Neural Network (ST-CNN)

The proposed diagnosis framework, spatiotemporal convolutional neural network (ST-CNN), consists of spatiotemporal feature learning using STPN and condition classification with CNNs.

3.1. Spatiotemporal feature learning with STPN

The STPN is applied to extract spatial (between measurements) and temporal features (for each measurement), which is built on the formulation of transition probabilities among the states generated by SDF. With the introduction of STPN in Section 2, the spatiotemporal feature learning process is illustrated as follows.The transition probabilities of states between the sub-systems represent the characteristics of the system, where the atomic pattern denotes the features within each individual subsystem (i.e., temporal features for the time-series) and the relational pattern implies the features between two different subsystems (i.e., spatial features for the multivariate time-series). With the transition probabilities, the spatiotemporal features can be extracted by STPN.For the symbolic systems shown in Fig. 1, four transition matrices are formed including two self-symbol generation matrices (Π^{aa}, Π^{bb}) and two cross-symbol generation matrices (Π^{ab}, Π^{ba}). The self-symbol generation matrix represents the transition probabilities in the symbol system itself, which can be computed using *D*-Markov machine. That's to say, the self-symbol generation matrices describe the characteristics of the system in terms of the temporal features. And the cross-symbol generation matrix is obtained via inspecting the transition probabilities between two symbol systems with x*D*-Markov machine (in fault diagnosis field, the two symbol systems can be two measurements of the mechanical system), which are intended to extract the spatial features between the symbol systems. It should be noted that the cross-symbol generation matrices between the systems a and b are not symmetric, i.e., Π^{ab} and Π^{ba} are not necessarily same.With this setup, both spatial and temporal features are extracted via the STPN model. The self-state transition matrices are expressed as:

$$\begin{aligned} \pi^{aa}_{\sigma_i\sigma_j} &\triangleq P(q^a_{k+D^{aa}} = \sigma_j | q^a_k = \sigma_i) \quad \forall k \\ \pi^{bb}_{\tau_i\tau_j} &\triangleq P(q^b_{k+D^{bb}} = \tau_j | q^b_k = \tau_i) \quad \forall k \end{aligned} \tag{3}$$

where $\sigma_i, \sigma_j \in Q^a, \tau_i, \tau_j \in Q^b$, and D^{aa}, D^{bb} are the depth of the *D*-Markov machine for symbol systems a and b respectively.

And the cross-state transition matrices are expressed as:

$$\begin{aligned} \pi^{ab}_{\sigma_i\tau_j} &\triangleq P(q^b_{k+D^{ab}} = \tau_j | q^a_k = \sigma_i) \quad \forall k \\ \pi^{ba}_{\tau_j\sigma_i} &\triangleq P(q^a_{k+D^{ba}} = \sigma_i | q^b_k = \tau_j) \quad \forall k \end{aligned} \tag{4}$$

where D^{ab}, D^{ba} are the depth of the x*D*-Markov machine for the transitions from symbol system a to symbol system b and the transitions from symbol system b to symbol system a respectively.

The transition probability π^{aa} in a self-state transition matrix can be obtained based on the formulation of the *D*-Markov machine: $\pi^{aa}_{\sigma_i\sigma_j} = N^{aa}_{\sigma_i\sigma_j} / N^{aa}_{\sigma_i}$, where $N^{aa}_{\sigma_i\sigma_j}$ is the number of times that the state $\sigma_j \in Q^a$ is emanated after the state $\sigma_i \in Q^a$, i.e.,

$$N^{ab}_{\sigma_i\sigma_j} \triangleq |\{(Q^a(k), Q^a(k+D^{aa})) : Q^a(k+D^{aa}) = \sigma_j \mid Q^a(k) = q^a_i\}| \tag{5}$$

where $Q^a(k)$ is the k^{th} state in the state sequence Q^a and $S^a(k+D^{aa})$ is the $(k+D^{aa})^{th}$ symbol in the state sequence Q^b, and $N_m = \sum_{j=1}^{|Q^a|} N^{aa}_{\sigma_i\sigma_j}$.

Similarly, the transition probability π^{bb} for symbolic system b can be computed.

For the transition probability π^{ab} in a cross-state transition matrix, π^{ab} is obtained using the xD-Markov machine: $\pi^{ab}_{\sigma_i\tau_j} = N^{ab}_{\sigma_i\tau_j}/N^{ab}_{\sigma_i}$, where $N^{ab}_{\sigma_i\tau_j}$ is the number of times the state $\tau_j \in Q^b$ is emanated after the state $\sigma_i \in Q^a$, i.e.,

$$N^{ab}_{\sigma_i\tau_j} \triangleq |\{(Q^a(k), Q^b(k+D^{ab})) : Q^b(k+D^{ab}) = \tau_j \mid Q^a(k) = q^a_i\}| \tag{6}$$

where $Q^a(k)$ is the k^{th} state in the state sequence $Q^a, Q^b(k+D^{ab})$ is the $(k+D^{ab})^{th}$ symbol in the state sequence Q^b, and $N_m = \sum_{j=1}^{|Q^b|} N^{ab}_{\sigma_i\tau_j}$.

Similarly, the transition probability π^{ba} can be computed. Note that, when the depth $D = 1$, the state transition probabilities are equal to the symbol generation probabilities [37].

With the STPN model, the extracted spatial and temporal features are represented by the cross-transition and self-transition probabilities (Eqs. (3) and (4)). For a state transition matrix Π^{ab}, the probabilities are in 2 dimensions, and they can be applied as the inputs for the CNN model. In this work, the number of bins is chosen as 6 with MEP approach, and the depth D is equal to 2. Therefore, the input size is 36×36 for the CNN.

An explanatory example is shown in Fig. 2 to illustrate the spatiotemporal feature extraction using the proposed approach.

3.2. Diagnosis framework with ST-CNN

The ST-CNN framework for multivariate time-series is shown in Fig. 3 including the spatiotemporal feature extraction with STPN (the top panel) and the classifier with CNNs (the bottom panel).

Based on the STPN model (Section 3.1), the spatiotemporal features are extracted and represented as 2D images (dimensions of rows and columns are both 36). For the multivariate time-series $X = \{X^{\mathbb{A}}(t), t \in \mathbb{N}, \mathbb{A} = 1, 2, \cdots, f\}$, f^2 channels are formed including the self-state transition matrices Π^{aa} and the cross-state transition matrices Π^{ab}. The generated transition matrices (converted into images) are applied as the input to the CNN model (the bottom-right panel of Fig. 3).

3.2.1. Network parameters

The data set is prepared in diverse operating conditions (case study 1 in Section 4) or fault severities (case study 2 in Section 4). The training data and testing data are formed via random sampling, where two testing scenarios are assumed: (i) the testing samples are with the same operating conditions or fault severities of the training data (noted as testing on trained OC or FS), and (ii) the testing samples are with different operating conditions or fault severities (noted as testing on untrained OC or FS). The later scenario is intended to evaluate the generalization ability of the diagnosis approach. For real industrial applications, it may be a hard work to collect the samples under various OCs or FSs for model training. Consequently, it is highly desirable to design the approaches that possess strong generalization ability and can learn the transferable features across different OCs and FSs. The number of samples for training and testing is listed in the results section for each data set. The training data is further splitted into training set and validation set with ratios 0.8 and 0.2 respectively.

The structure and parameters of the ST-CNN are listed in Table 1.

Three convolutional layers are applied to learn the features from STPN, where filter size is 3×3 for each convolutional layer and the number of filters ranges from 16 to 1024. Three subsampling layers are applied after each convolutional layer

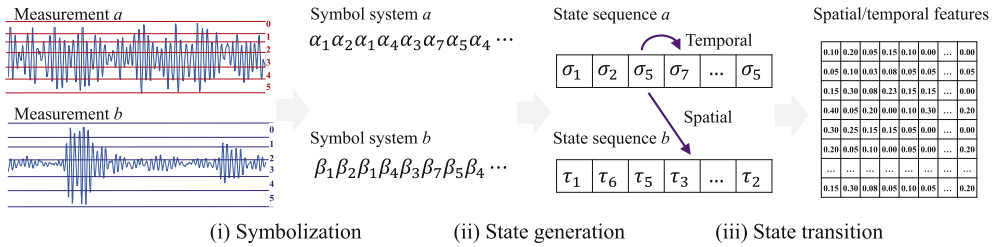


Fig. 2. The spatiotemporal features extracted by the proposed approach. Three steps are included, (i) the measurements are first partitioned into the symbol systems through symbolization process, (ii) the state sequences are generation via the predefined depth D, and (iii) the state transition probabilities are calculated using the formation of D-Markov machine and xD-Markov machine. Here, the spatial features are represented by the transition probabilities between the two systems and the temporal features are extracted within a system which indicates the state transition characteristics itself. For each state transition matrix $\Pi_{ij}, i, j = a, b$, a 2-D matrix is formed and $f^2 = 4$ matrices are formed in this example as the number of time-series f equals to 2.

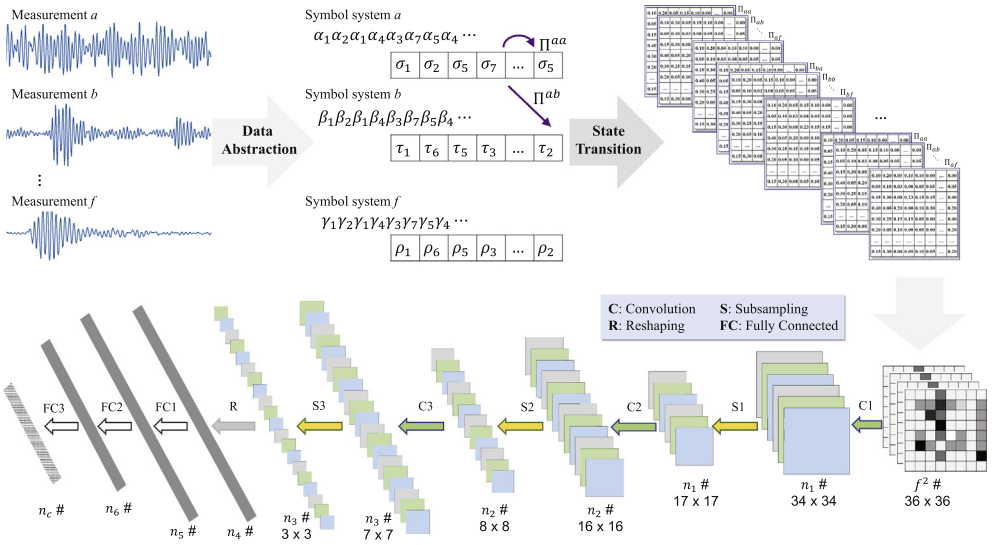


Fig. 3. The proposed ST-CNN framework consisting of spatiotemporal feature extraction via STPN and condition classification via CNNs.

Table 1
Model structure of ST-CNN.

Layer name	Filter number	Filter size	Activation
conv1 (C1)	n_1	3×3	–
max-pool1 (S1)	–	2×2	RELU
conv2 (C2)	n_2	3×3	–
max-pool2 (S2)	–	2×2	RELU
conv3 (C3)	n_3	3×3	–
max-pool3 (S3)	–	2×2	RELU
Flatten (R)	–	–	–
FC1	n_4	–	RELU
FC2	n_5	–	RELU
FC3	n_6	–	RELU
N	n_c	–	Softmax

using max pool function with pooling size 2×2. Batch normalization and dropout (ranging from 0.1 to 0.6) are applied with ReLU activations. The fully-connected layers comprise 3 layers of 128 to 1024 hidden units each and trained with dropout fractions of 0.1 to 0.6 using ReLU activations. The classification layer is finally formed with Softmax activation function (the bottom-left panel in Fig. 3). A batch size ranging from 8 to 256 is used. All hyperparameters as described are chosen by cross-validation, and the set of optimal hyperparameters varies depending on the data set. The training procedure employs the early-stopping algorithm (with patience 3) where training stops when validation error ceases to decrease.

3.2.2. Remark

The conversion of the spatiotemporal features (from STPN) to the inputs of CNNs can be single-channel or multi-channel in terms of the input shape for the CNNs. The structure shown in Fig. 3 is multi-channel, where the number of channels is f^2 and the dimension for each channel is 36×36. For the single-channel setting, the features from each atomic pattern (temporal features) or relational pattern (spatial features) can form one image (with dimension $f \cdot 36 \times f \cdot 36$ in this case). Also, the dimension for each pattern depends on the selection of the parameters ($|Q|$–the number of states and D-the depth of D–Markov machine and xD–Markov machine).

3.3. Comparison methods

The comparison methods applied in this work include deep learning approach and shallow methods. For the deep learning approach, CNN shows exceptional performance in fault diagnosis with time-series [24], and is adopted as the baseline method, where a 1D CNN structure with multiple input channels is formed in this work for multivariate time-series. Among the shallow methods, Support Vector Machine and Random Forest are widely applied and used for comparison.

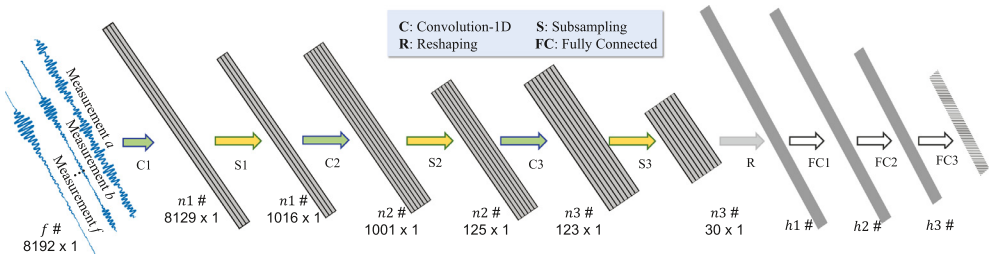


Fig. 4. 1D CNN model for fault diagnosis with multivariate time-series.

3.3.1. 1D CNN

In image processing, a 2D structure is widely employed for the reasons of natural 2D space correlation in images. A similar way, which convert 1D time-series to spectrogram images for CNN training, is also adopted in the fault diagnosis of mechanical systems. More generally, many studies use a 1D convolutional structure to process 1D mechanical signal, and they hold the opinion that the features can be easier to be grasped and extracted in its original dimension [14].

By introducing multiple channels, the 1D CNN model can be applied in multivariate cases, and the framework is shown in Fig. 4.

3.3.1.1. Network parameters. The data set is same as that used for ST-CNN as well as the training and testing samples (as discussed in Section 3.2). The structure and parameters of the 1D CNN are listed in Table 2.

Three convolutional layers are applied to learn the features from multivariate time-series inputs, where filter sizes are 64, 16, and 3 respectively (as suggested in [24], wider filter size is adopted in the first two convolutional layers) and the number of filters ranges from 16 to 1024. Three subsampling layers are applied after each convolutional layer using max pool function with pooling sizes 8, 8, and 4 respectively. Batch normalization and dropout (ranging from 0.1 to 0.6) are applied with ReLU activations. The fully-connected layers comprise 3 layers of 128 to 1024 hidden units each and trained with dropout fractions of 0.1 to 0.6 using ReLU activations. The classification layer is finally formed with Softmax activation function. A batch size ranging from 8 to 256 is used. The optimal architecture (the number of the convolutional layers and the fully connected layers) is determined by employing trial-and-error strategy as well as the hyperparameters. The Stochastic Gradient Descent (SGD) optimizer is applied in the training process. The training procedure also employs the early-stopping algorithm (with patience equal to 3).

3.3.2. Shallow methods

Support Vector Machine (SVM) and Random Forest (RF) are applied in this work as the comparison methods. SVM and RF are two typical shallow methods which have been shown to be efficient and useful in kinds of fault diagnosis applications [44–46]. A brief description of the two methods is given as follows for clarity.

3.3.2.1. SVM. The basic idea of the SVM model is to create an optimal hyperplane, that can guarantee the largest distance to the closest training-data point of any class, for classification or regression tasks. The radial basis function (RBF) kernel is adopted in this work, and the hyperparameters are selected using cross-validation strategy on the training set.

3.3.2.2. Random Forest. In this work, the only two structure parameters, namely the number of trees in the forest n_{try} and the number of random feature subset m_{try}, are set to 500 and $|\sqrt{m}|$, respectively, where m is the dimension of input feature vector.It should be noted that the above shallow methods cannot learn feature from raw signal directly, and manual feature

Table 2
Model structure of 1D CNN.

Layer name	Filter number	Filter size	Activation
conv1 (C1)	n_1	64×1	–
max-pool1 (S1)	–	8×1	RELU
conv2 (C2)	n_2	16×1	–
max-pool2 (S2)	–	8×1	RELU
conv3 (C3)	n_3	3×1	–
max-pool3 (S3)	–	4×1	RELU
Flatten (R)	–	–	–
FC1	h_1	–	RELU
FC2	h_2	–	RELU
FC3	h_3	–	RELU
N	n_c	–	Softmax

extraction is necessary before model training and diagnosis. In this work, two kinds of features are applied, (i) statistical features consists of 16 time-domain features (mean, root mean square, square-root, absolute mean, skewness, kurtosis, max, min, peak-to-peak, variance, waveform index, peak index, impulse factor, tolerance index, skewness index, and kurtosis index) and 13 frequency-domain features (defined in Table II [19]), (ii) the time–frequency features extracted by wavelet packet analysis, where the vibration signal is decomposed by wavelet packet analysis using the db2 basis and the decompose level equal to 3 and then the statistical features from each component are obtained.

3.4. Computational complexity

3.4.1. ST-CNN

For the spatiotemporal feature learning process (with a fixed STPN structure–the number of symbols and depth for D-Markov and xD-Markov machines are fixed), the computational cost is $\mathcal{O}(f^2T)$ (based on Eqs. (3) and (4), the complexity for each symbol generation matrix is $\mathcal{O}(T)$, and there are f^2 matrices in the STPN model). Here, T is the length of each input sample. The computational complexity for the convolutional layers of a 2D CNN is [47,48]: $\mathcal{O}(\sum_{i=1}^{N} n_{i-1} \cdot s_i^2 \cdot n_i \cdot m_i^2)$ where i is the ith convolutional layer, N is the number of convolutional layers, n_i is the number of filters (noted as width) in the ith layer, n_{i-1} is the number of the input channels of the ith layers (is also the number of layers of the $(i-1)th$ layer), s_i is the spatial size of the filter (noted as length), m_i is the spatial size of the output feature map. For the fully-connected layers and the pooling layers, the time consumption is usually 5–10% of the time consumed by the convolutional layers [47,48]. Thus, the overall computational cost for ST-CNN is:

$$\mathcal{O}(f^2T + k\sum_{i=1}^{N} n_{i-1} \cdot s_i^2 \cdot n_i \cdot m_i^2)$$

where k is the factor ($k = 1.05 - 1.1$ when considering the computational cost of the fully-connected layers and the pooling layers of CNN).

3.4.2. 1D CNN

The overall computational cost for 1D CNN is:

$$\mathcal{O}(k\sum_{i=1}^{N} n_{i-1} \cdot \mathbb{s}_i \cdot \mathbb{m}_i \cdot \mathbb{m}_i)$$

As discussed in [24], the filter size of the first convolutional layer should be much longer than that used in 2D CNN, $\mathbb{s}_1 = \mathbb{64}$ or even longer is suggested to reduce the computational cost.

For the ST-CNN and 1D CNN with similar structures (the number of layers, number of filters, and number of fully-connected layers), we have,

$$\mathcal{O}(f^2T + k\sum_{i=1}^{N} n_{i-1} \cdot s_i^2 \cdot n_i \cdot m_i^2) \ll \mathcal{O}(k\sum_{i=1}^{N} n_{i-1} \cdot \mathbb{s}_i \cdot \mathbb{m}_i \cdot \mathbb{m}_i)$$

The reason is that, we have $\mathbb{m}_i \approx \mathbb{T} \gg \mathbb{m}_i^2$, where $T = 8192$ in this work, $\mathbb{s}_1 \gg \mathbb{s}_1^2$, and this means that the computational cost of the first convolutional layer of 1D CNN is far more than the ST-CNN. For the remaining layers, although $\mathbb{s}_i \approx \mathbb{s}_i$ for $i > 1$, the decrease of m_i is p^2 times faster than $\mathbb{m}_i$ (suppose the pooling size is p for both models). If $\mathbb{s}_i = \mathbb{2}, \mathbb{s}_i = \mathbb{2}, \mathbb{p} = \mathbb{2}$ (these are the most usually used hyperparameters), the decreasing speed of the computational cost of the remaining layers (from the second layer) is same. Therefore, the computational cost of 1D CNN is far more than ST-CNN. Note that the filter size and pooling size may vary in the searching of optimal hyperparameters, while in most cases the computational time of ST-CNN is far less than 1D CNN for both case studies in this work. Detailed comparison on the computation time will be explained in Section 4.3.

It should also be noted that the computational costs of SVM and RF are less than ST-CNN and 1D CNN, as they are shallow models with fewer parameters to be trained.

4. Results and discussions

4.1. Case study on diagnosing untrained operating conditions with wind turbine data

4.1.1. Data set on wind turbine test rig

The first dataset is from our wind turbine test rig. The detailed schematic diagram of experimental platform is shown in Fig. 5. It mainly consists of wind tunnel, direct-drive wind turbine test bench, accumulator, and signal acquisition system. To create a more realistic running environment of wind turbine, we use the wind tunnel to generate the wind sources and drive the direct-drive wind turbine instead of motor drive in several existing experiments. This scheme contributes to the simulation of a comprehensive mechatronics system in real wind turbine. Multiple sensors are placed in diverse positions to

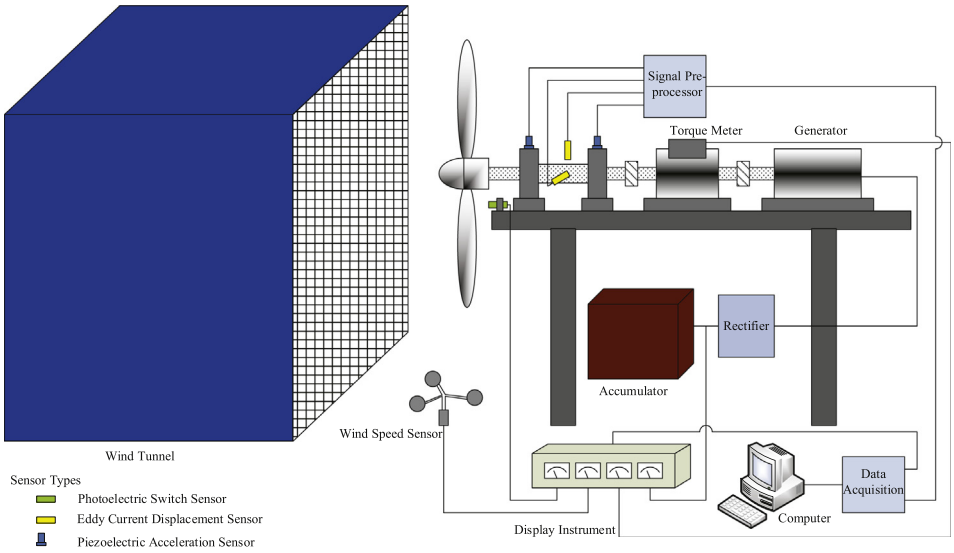


Fig. 5. The schematic diagram of experimental platform.

Table 3
Variables measured in the experiments on wind turbine test rig.

ID	Variable
1	Wind speed (m/s)
2	Rotor speed (rpm)
3	Torque (Nm)
4	Wind power (W)
5	Displacement (horizontal, μm)
6	Displacement (vertical, μm)
7	Acceleration (front bearing, g)
8	Acceleration (back bearing, g)

Table 4
Tested conditions (normal and 11 faults) on wind turbine test rig.

Condition ID	Description
1	Normal
2	Misalignment, horizontal direction
3	Misalignment, vertical direction
4	Loosening, front bearing support
5	Loosening, back bearing support
6	Rolling element fault, front bearing
7	Inner-ring fault, front bearing
8	Outer-ring fault, front bearing
9	Mass unbalance of wind wheel
10	Variation in airfoil of blades
11	Yaw fault
12	Aero-asymmetry of wind wheel

monitor eight variables, which are listed in Table 3. The sampling frequency is 20 kHz. Twelve machine conditions are simulated in test rig, which are listed in Table 4. All these faults can fundamentally cover the common failure modes from wind wheel to drivechain.

Considering the varying operating conditions for real wind turbine, we perform the experiments in six different loading conditions, as listed in Table 5. It should be noted that the values in Table 5 are the averaged ones over a period of sampling time. Both the wind speed and rotating speed of wind wheel are slightly fluctuant over time in each loading condition and it accords with the actual operating conditions. The data is prepared with segments with 8192 time steps for each operating

Table 5
Description of multiple operating conditions.

Loading condition	Wind speed (m/s)	Rotor speed (rpm)
Load 1	5.8	255
Load 2	6.9	260
Load 3	8	267
Load 4	9.2	276
Load 5	10.3	288
Load 6	11.5	300

condition. In this case study, two scenarios are formed, (i) the loads 1–3 are defined as seen OCs, while loads 4–6 are unseen OCs, and (ii) the loads 4–6 are defined as seen OCs, while loads 1–3 are unseen OCs. And the averaged accuracies are obtained based the two scenarios.

The wind turbine can be treated as a complex system where the rotor, bearing, generator, controller, tower, yaw controller, and wind wheel are subsystems of it. Although the test rig is greatly simplified, several subsystems can still be separated such as wind wheel, rotor, bearing, and generator. As a result, the faults occurred in wind wheel (e.g., unbalance of wind wheel, variation of blades' airfoil) are more likely to be detected by the measurements that represent the subsystem's behavior such as wind speed-power curve, and are more difficult to be diagnosed via the measurements from the generator or bearing. In this context, 11 faults are separated into two categories by generally associating the faults to their locations: (i) conditions 2–8 (listed in Table 4) which are the typical faults attributed to the drivetrain including the rotors, bearing, and bearing support, and (ii) condition 9–12 and these faults are located at the wind wheel. As discussed above, the classification of conditions 9–12 is more accurate when incorporating the wind speed-power curve, which is the relationship between the wind speed and power output. Similarly, the relationship between the wind speed, rotor speed, torque of drivetrain, and power output should also help the classification and these are the spatial features between the variables 1–4 listed in Table 3. Therefore, the parameters are separated into two categories: (i) variables 1–4 (listed in Table 3) that indicate the performance of the wind turbine and (ii) variables 5–8 (Table 3) which are vibration measurements and mostly represent the state of the drivetrain subsystem. Although the variables 5–8 are all installed within the drivetrain subsystem, the accelerators (variables 7 and 8) are more applied in fault diagnosis of bearing faults. With this setup, three data sets are formed and listed in Table 6.

4.1.1.1. Feature sets for shallow methods. As discussed in Section 3.3.2, two kinds of features are applied, (i) statistical features extracted from the raw time-series, and (ii) statistical features extracted from the decomposed time-series using wavelet packet analysis. As a result, two feature sets are formed in this case, which are listed in Table 7.

Note that the two feature sets are formed based on different understanding of the system. Although wavelet packet analysis is well known and it is becoming one of the classical approaches in feature extraction, there are still parameters that depends on the understanding of the signal and the system such as the mother wavelet, the number of the components to be extracted, etc. And the performance of the diagnosis heavily relies on these parameters.

4.1.2. Performance on fault diagnosis in diverse operating conditions

For the three configurations listed in Table 6, the classification accuracy using the proposed approach and the comparison methods are listed in Tables 8, 9 and 10.

The first configuration 'WT2-8' only includes the faults that are located at the drivetrain, and 1D CNN presents the best accuracy for both the testing on the seen OCs and the unseen OCs. Note that the testing on seen OCs means that the testing samples are from the same OCs as training, but different from the training samples. And the testing on unseen OCs means that the testing samples are with different OCs in comparison to the training data. For the shallow methods, the performance is good for the seen OCs (87.7% and 89.4% for SVM-1 and RF-1 respectively, 98.5% and 97.3% for SVM-2 and RF-2 respectively), while the accuracy drop is obvious on the unseen OCs. In this context, 1D CNN and ST-CNN outperform SVM and RF. It should be noted that ST-CNN does not perform as well as 1D CNN on the unseen OCs. The reason is that, the data abstraction of STPN loses some of the information that might be helpful for the classification and further analysis is being carried out to adjust the abstraction process while preserving the ability of adaptive feature learning.

When the number of conditions increases and the faults from the wind wheel are included, the diagnosis accuracy is significantly decreased, especially for the unseen OCs. In this case, 1D CNN still obtained a high accuracy on the seen OCs

Table 6
Configurations of wind turbine data set used for comparison.

Data set	Num of variables	Variable IDs	Num of conds	Cond. IDs
WT2–8	2	7, 8	8	1:8
WT2-12	2	7, 8	12	1:12
WT8-12	8	1:8	12	1:12

Table 7
Feature sets used for shallow methods (SVM and RF) in wind turbine data set.

Feature set	Number of features			Description
	WT2–8	WT2-12	WT8-12	
1	58	58	232	Statistical features for each variable[a]
2	464	464	1044[b]	Features of each component obtained by wavelet packet analysis

[a] The statistical features consists of 16 time-domain features (mean, root mean square, square-root, absolute mean, skewness, kurtosis, max, min, peak-to-peak, variance, waveform index, peak index, impulse factor, tolerance index, skewness index, and kurtosis index) and 13 frequency-domain features (defined in Table II [19]). The same feature extraction setting is applied on the decomposed signal for feature set 2.
[b] The wavelet packet analysis is implemented on the 4 vibration signals (displacement and acceleration) using the db2 basis and the decompose level equal to 3, where 928 features formed. The features of the other 4 variables (wind speed, rotor speed, torque, wind power) are same as feature set 1.

Table 8
Performance of 1D CNN and ST-CNN on wind turbine data set.

Data set	Num of variables	Num of conds.	Training samples	Testing samples	Accuracy	
					1D CNN	ST-CNN
WT2-8	2[a]	8[a]	40064	7071/ 7071[b]	100.0/ 99.4[c]	98.6/ 94.0
WT2-12	2	12	60098	10606/ 10606	96.8/ 72.6	88.2/ 80.5
WT8-12	8	12	60098	10606/ 10606	97.4/ 78.4	99.0/ 90.9

[a] The used variables and conditions can refer to Table 6.
[b] The testing samples are from seen and unseen OCs respectively.
[c] The accuracies are with testing samples on seen and unseen OCs respectively. And the average accuracy is shown here based on the two scenarios, (i) the loads 1–3 are defined as seen OCs, while loads 4–6 are unseen OCs, and (ii) the loads 4–6 are defined as seen OCs, while loads 1–3 are unseen OCs.

Table 9
Performance of SVM and Random Forest on wind turbine data set with feature set 1.

Data set	Num of variables	Num of conds.	Training samples	Testing samples	Accuracy	
					SVM	RF
WT2-8	2	8	800	800/ 800	95.7/ 87.7	99.1/ 89.4
WT2-12	2	12	800	800/ 800	91.0/ 67.5	96.7/ 70.3
WT8-12	8	12	800	800/ 800	95.8/ 73.1	98.8/ 74.2

Table 10
Performance of SVM and Random Forest on wind turbine data set with feature set 2.

Data set	Num of variables	Num of conds.	Training samples	Testing samples	Accuracy	
					SVM	RF
WT2–8	2	8	800	800/ 800	99.1/ 98.5	100.0/ 97.3
WT2-12	2	12	800	800/ 800	93.7/ 76.0	97.7/ 81.7
WT8-12	8	12	800	800/ 800	98.9/ 83.5	99.9/ 84.2

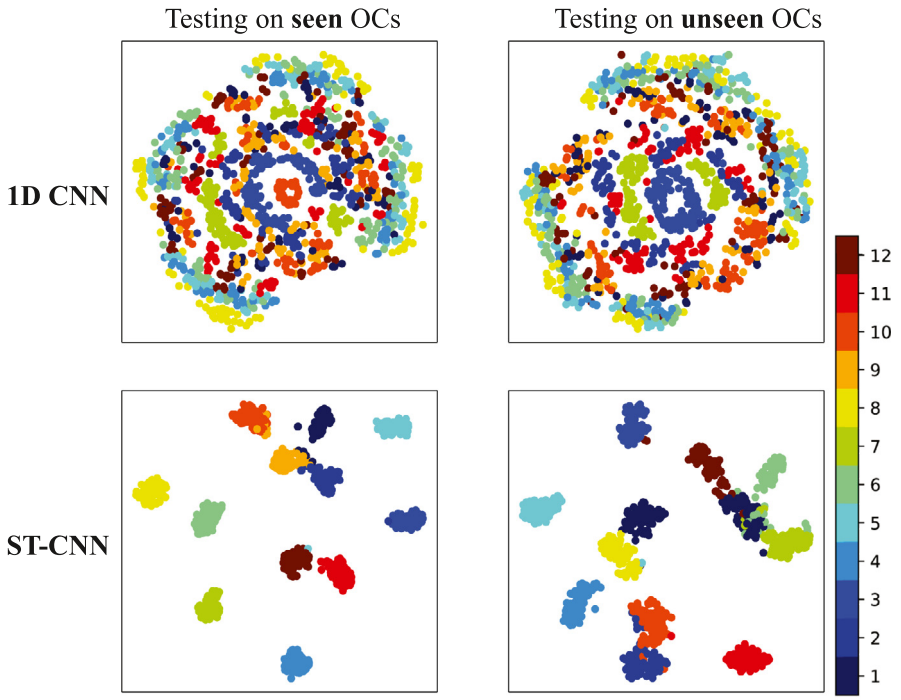
(96.8%), which proves the extraordinary fitting power of CNNs. However, the probably overfitted model fails to diagnose the unseen OCs (with accuracy lowered to 72.6%). And the gap between the seen and unseen OCs for ST-CNN is lower than that of 1D CNN, which indicates that the features learnt by STPN are more transferable between different OCs.

With the additional information (8 variables) used, the ST-CNN outperforms the other methods. This shows that the proposed framework is capable of dealing with multivariate time-series and efficiently extracts the features for classification.

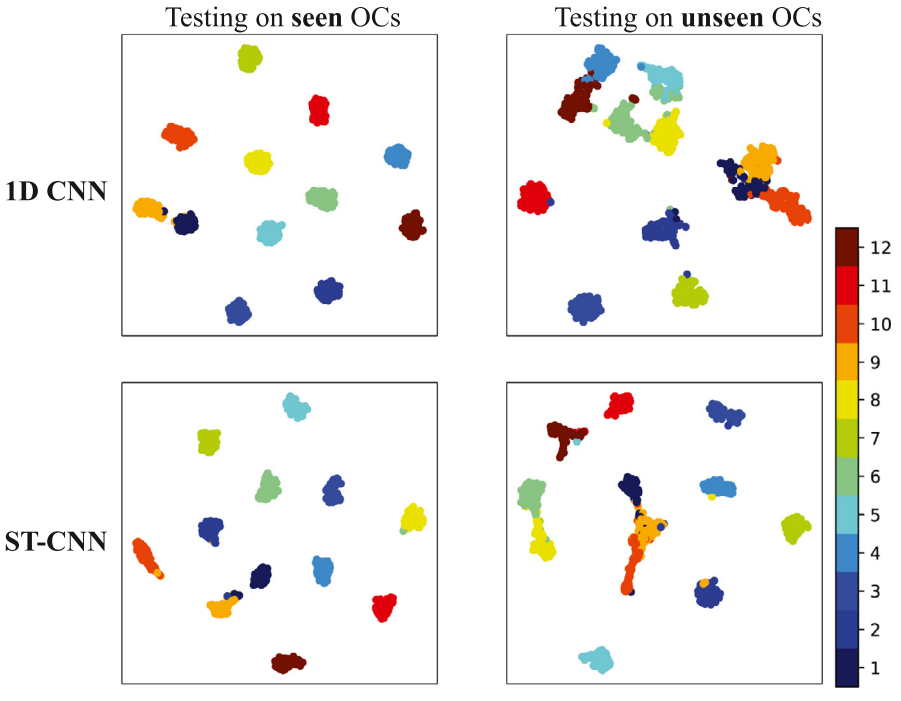
It should be noted that the number of training samples for SVM and RF are much less than that for CNN and ST-CNN. The reason is that the shallow methods (SVM and RF in this work) can achieve a good performance under small sample size from existing works.

4.1.3. Spatial features learnt by STPN

The results show that the ST-CNN outperforms 1D CNN on data set WT8-12, where 8 variables are applied for classifying 12 conditions. One of the possible reason is that the 1D CNN fails to extract the useful features in complex systems with different types of data. To validate this, t-Distributed Stochastic Neighbor Embedding (t-SNE) [49] is applied to visualize the features in different levels of the models which are shown in Fig. 6, including the features of the last convolutional layer (Fig. 6), and the features of the last fully-connected layer (Fig. 6(b)).



(a) Convolutional layer (last)



(b) Output layer

Fig. 6. Features leant by ST-CNN and 1D CNN. Data set WT8-12 is used and t-SNE is applied to obtain the 2 dimensional embedding from the high dimensional feature space. The testing on seen operating conditions (OCs) and unseen OCs are shown respectively.

The features of ST-CNN at the last convolutional layer are more classifiable than that of 1D CNN (shown in Fig. 6(a)), which means that the features extracted by STPN properly represent the system characteristics and aid the CNN model to form more separable states. The 1D CNN model finally gets well-represented features at the last fully-connected layer (Fig. 6(b)) where most of the conditions can be separated. However, there are still several conditions close to each other on the unseen OCs (top-right panel of Fig. 6(b)), where conditions 1, 9 and 10 and conditions 4 and 12 are overlapping. This can also be observed from the confusion matrices (shown in Fig. 7a). For the ST-CNN, the features of the last fully-connected layer are further classifiable for both the seen and unseen OCs, although there are also overlapping for the unseen OCs (conditions 9 and 10 shown in the bottom-right panel of Fig. 6(b), and this leads to the misclassification of condition 10 to condition 9 as observed from the confusion matrix in Fig. 7b).

By comparisons with the misclassified conditions shown in the confusion matrices (Fig. 7), the 1D CNN model fails to distinguish the conditions 9 and 10 from normal condition, which is critical as these true negatives would detect the faults as normal and may cause serious consequences. It can also be observed that the worst-case accuracy of ST-CNN is higher than 1D CNN. As discussed above, the fault 9–12 are related to the wind wheel, which is supposed to be detected by the wind speed-power curve (spatial features between wind speed and power output). The results here show that the 1D CNN does not properly capture the spatial features between the wind speed, rotor speed, torque, and power output.

To address the question that if the spatial features learnt by STPN can assist the classification of the aforementioned conditions, t-SNE results of three condition sets (set 1–conditions 1 and 9, set 2–conditions 1 and 10, set 3–conditions 4 and 12, these condition sets are mostly misclassified by 1D CNN as shown in Fig. 7a) are shown in Fig. 8 where the relational patterns (relationship between variables) are applied as the features. It can be observed that the listed conditions can be generally separated by the spatial features listed below each subfigure.

In the condition 9 (mass unbalance of wind wheel), one blade is with additional mass, and this influence the relationship between the wind speed and the rotor speed (also between the wind speed and power output). Two relational patterns (Π_{12}, and Π_{14}) are used to identify the fault. For the condition 10, the variation in airfoil of blades changes the power coefficient C_p and this affects the energy conversion (i.e., wind speed-power curve). The variation of power output also alters

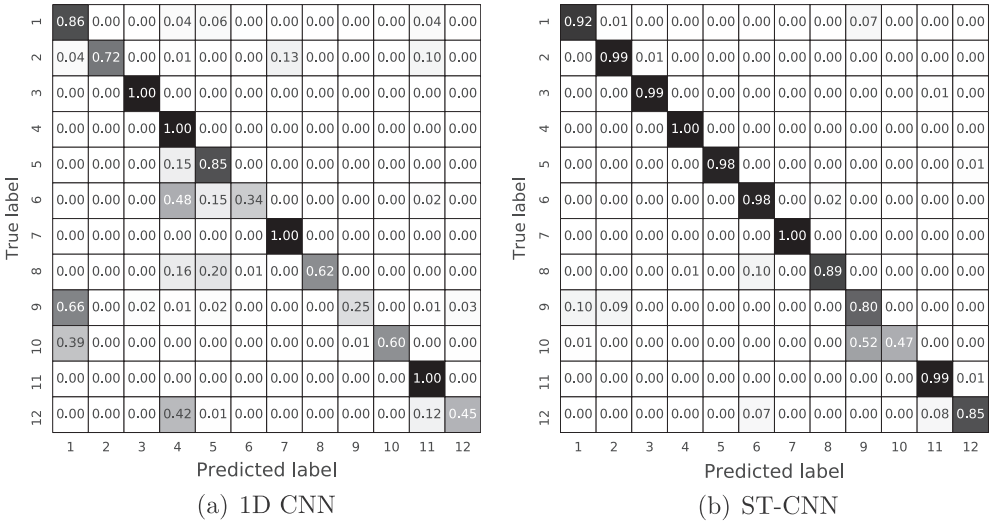


Fig. 7. Confusion matrices with data set WT8-12.

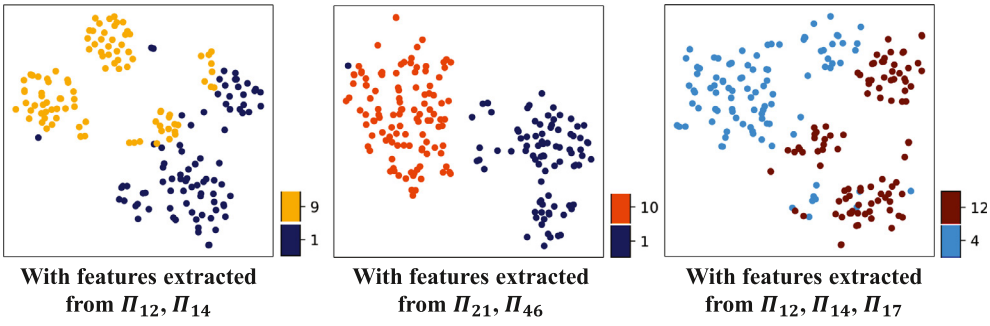


Fig. 8. Visualization (via t-SNE) of spatial features learnt by STPN.

the relationship between the power output and the vibrations (e.g., Π_{46}). Here, two relational patterns (Π_{21} and Π_{46}) are chosen to distinguish the condition 10 from normal condition 1. Similarly, condition 12 (aero-asymmetry of wind wheel) changes the airfoil of one blade which also influences the power coefficient, and three relational patterns (Π_{12}, Π_{14}, and Π_{17}) are selected to separate the conditions 12 and 4.

Using the spatial features (relational patterns) extracted by STPN, the faults associated with the variation of wind speed-power curve are separable (via t-SNE). This validates that the spatial features are properly represented by the STPN model and can be used for condition classification. It should be noted that the above mentioned spatial features might not classify conditions 1, 9, 10 and 12 simultaneously, as the variation of the wind speed-power curve all exists in conditions 9, 10, and 12, and more information is needed to separate them.

4.1.4. Spatiotemporal features activated for classification

In the proposed framework, STPN learns both spatial and temporal features from multivariate time-series and the CNN model makes the decision on classification. Fig. 8 shows that the spatial features provide the information to separate some of the conditions. However, the question that which (spatial or temporal) features are applied for the classification should be addressed to understand the classifier. Through this, we can also get more sense that why this condition is correctly classified or not. This work adopts the Class Activation Mapping (CAM) and Gradient-weighted Class Activation Mapping (Grad-CAM) [50,51] to compare the activations on spatiotemporal features as shown in Fig. 9.

From the activation map of spatiotemporal features, the questions that which spatial or temporal features are applied for classification and which variables are more useful in terms of distinguishing one condition from the others can be addressed. In some of the conditions (e.g., 4 and 5), the temporal features (along with the diagonal blocks of the subfigure) are mostly used, while in most of the conditions, both spatial and temporal features contribute on the decision. For instance, the temporal features Π_{77} (the acceleration of the front bearing) shows that they are more important than the other features in separating the condition 4 (the loosening of the front bearing), and this is consistent with the reality that the loosening of the front bearing affects the vibration of itself. For the yaw fault (condition 11), the rotating speed of the rotor is lowered, and the relationship between the rotor speed and the vertical displacement of the rotor (i.e., the relational pattern Π_{26}) is different from the others as well as the relationship between the wind speed and the vertical displacement of the rotor (i.e., the relational pattern Π_{16}). For the conditions 9, 10 and 12 (discussed in Fig. 8), the spatial features are mostly applied for the classification. Although the activated features are not same as the ones shown in Fig. 8, the activated features here also works for the classification (if the spatiotemporal features in these conditions are masked with the activation map shown in Fig. 9, the conditions are completely classifiable via t-SNE).

It should be noted that the activation heatmap is not necessarily unique, and it depends on the model learnt from the data and the training strategy, considering the fact that the deep structure might not converge to the global optimum but probably a local one [52].

Based on the activation heatmap, the activated features can be further identified via Grad-CAM, and then the reasoning for the failed case in ST-CNN can be explained as well as the successful cases (while 1D CNN fails). Four cases are shown in Fig. 10, including three cases that are correctly classified (conditions 1, 9 and 10) and one failed case that misclassifies the condition 10 to condition 9.

Comparing with the cases (a) and (c) shown in Figs. 10a and 10c, the case (b) in Fig. 10b show that the activated features by ST-CNN is more similar to that of condition 9, not condition 10. One possible reason is that the STPN does not properly capture the relationship between the power output and the rotor speed (i.e., Π_{42}). This spatial feature is activated for classifying condition 10 (case (a)), while it is not activated in the case (b). By inspecting the raw data, the ratio of the power to

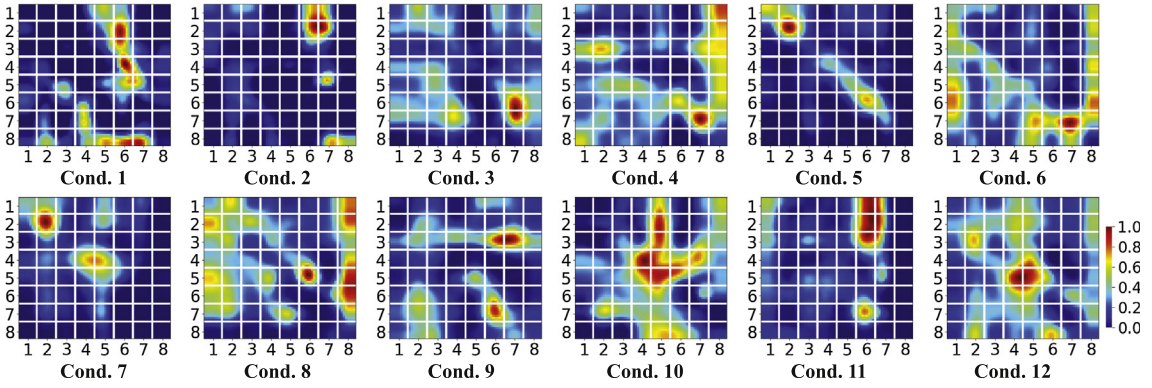
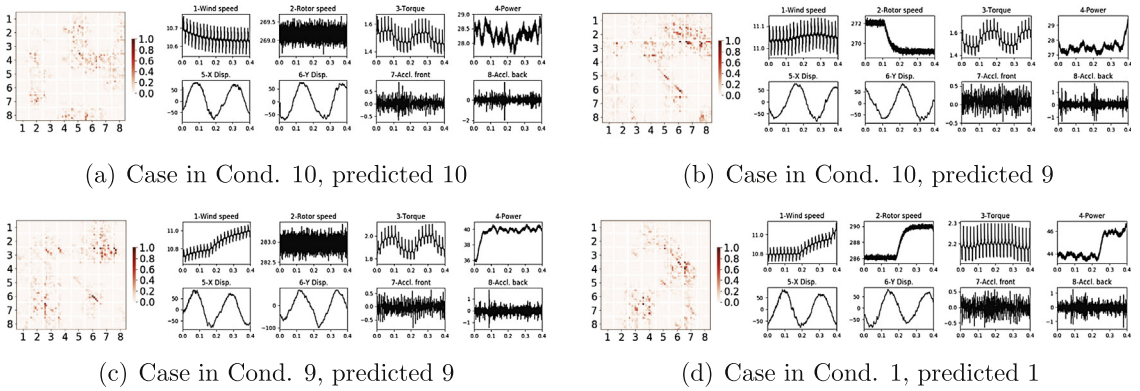


Fig. 9. Activation heatmap of spatiotemporal features learnt by STPN. The heatmap is obtained via CAM to signify the important zones that are used for classification, where larger value (red in the colormap) indicates that the region is activated and used for classifying the corresponding condition (listed below each subfigure). The numbers shown in the figure corresponding to the variable ID listed in Table 3, and the color block indexed by the y-axis ID i and x-axis ID j is the feature set represented by the state transition matrix Π_{ij} (spatial features for $i \neq j$, and temporal features for $i = j$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Case in Cond. 10, predicted 10

(b) Case in Cond. 10, predicted 9

(c) Case in Cond. 9, predicted 9

(d) Case in Cond. 1, predicted 1

Fig. 10. Cases on misclassified conditions of ST-CNN and 1D CNN for explaining how ST-CNN succeeds or fails.

the rotor speed of case (b) (0.16) is more close to that of case (c) (0.14), while it is far from that of case (a) (0.10), this might be the reason that the case (b) fails to be identified as condition 10. Compared with case (c), the case (b) presents similar activated zones (e.g., Π_{66}, Π_{37}, and Π_{38}), and this may let the case (b) be classified as condition 9. Further analysis is being carried out to interpret more failed cases and find the possible solution to improve the performance.

From the confusion matrices of 1D-CNN and ST-CNN (Fig. 7), 66% and 39% of the cases in conditions 9 and 10 are misclassified into condition 1 respectively, while the ST-CNN model mostly avoids such true negatives (the misclassification of conditions 9 and 10 for ST-CNN is discussed above). Here, comparing with the cases (a) and (c), the case (d) in condition 1 presents obvious differences on the activation map, where the spatial features (e.g., $\Pi_{36}, \Pi_{47}, \Pi_{46}, \Pi_{64}, \Pi_{73}$, and Π_{74}) are used to represent the normal condition, and these features imply the system characteristics in diverse OCs (that the absolute values for individual variables vary with OCs, while the relationship between them holds). In this context, the ST-CNN is more adaptive for diverse OCs by exploring the spatial features.

4.2. Case study on diagnosing unseen fault severity with bearing data set

To further validate the proposed framework for diagnosing unseen fault severities, the bearing data set collected by Case Western Reserve University [53] is used in this case.

4.2.1. Description of the data set

This bearing fault dataset has been widely used to verify diagnosis algorithm in numerous studies. The data set contains samples with four different bearing conditions, i.e., normal, inner race fault, roller fault, and outer race fault, under four motor speeds (1797 rpm, 1772 rpm, 1750 rpm and 1730 rpm). The vibration signal was collected by accelerometer with a sampling frequency of 48 kHz. For each fault type, three fault diameters (0.18 mm, 0.36 mm and 0.53 mm) were introduced by using electro-discharge machining. In this case study, we use the samples with 0.18 mm fault severity for training, while the samples with the other two severities (0.36 mm, 0.53 mm) for testing (also referred as unseen fault severities).

4.2.2. Performance on diagnosing untrained fault severities

With the data prepared in the above section, the performance of the proposed approach is tested and the results are listed in Table 12 as well as the comparisons with 1D CNN, SVM, and RF. To further improve the classification accuracy, an approach 'ST-CNN w SF' that incorporates both spatiotemporal features and statistical features (SF) in the time domain and frequency is presented and the results are also listed in Table 12, as the statistical features for acceleration have been well researched and these features can be obtained at (almost) no cost.

Using similar setting for the wind turbine data, two feature sets are formed for the shallow methods which are listed in Table 11.

Table 11
Feature sets used for shallow methods (SVM and RF) in bearing data set.

Feature set	Number of features	Description
1	29	Statistical features for each variable[a]
2	232	Features of each component obtained by wavelet packet analysis[b]

[a] The statistical features are same as that used in Table 7.
[b] The wavelet packet analysis is implemented using the db2 basis and the decompose level equal to 3.

For the samples with fault severity that are same as the training data, the classification accuracy for all of the methods is high. However, the performance of the samples with unseen fault severities is significantly reduced for the shallow methods (64.8% and 67.7% for SVM-1 and RF-1 respectively, 72.2% and 75.7% for SVM-2 and RF-2 respectively) and 1D CNN (73.6%). The ST-CNN and ST-CNN w SF outperform the other methods (90.6% and 92.4% respectively). Based on the ST-CNN framework, the incorporation of statistical features further improves the diagnosis performance, and this shows that proposed framework can efficiently fuse diverse information sources.

With statistical features (time domain and frequency domain) for the shallow methods, the trained model fits the situation with the same fault severity. When the fault severity varies, the statistical features change and the model cannot adjust the situation which results in the accuracy gap between the seen fault severity and unseen fault severities. The 1D CNN model also tries to fit the features in the seen fault severity and fails to learn the transferable features in different fault severities. For the ST-CNN, the features are represented by the transition probabilities which are more stable in different fault severities. Therefore, compared to 1D CNN and shallow methods, the proposed ST-CNN framework is more adaptive on unseen fault severities.

4.3. Discussions

Using the wind turbine data set 'WT8-12', the computational time for 1D CNN and ST-CNN is compared and listed in Table 13.

With the two case studies in Sections 4.1 and 4.2, the proposed ST-CNN framework: (i) can extract both the spatial and temporal features from multivariate time-series, (ii) is adaptive in handling diverse types of data and diagnosing unseen operating conditions and unseen fault severities, (iii) has no need of handcrafted features (although it can incorporate these features), (iv) is computational efficient (comparing to 1D CNN in Section 3.4), and (v) outperforms the shallow methods (SVM and RF) and the 1D CNN approach.

Signal processing techniques such as FFT and wavelet transform are well researched for several kind of measurements like acceleration, displacement, current and voltage. These approaches can be leveraged for feature extraction, but there may be requirements of manual steps in implementing these pre-processing techniques, such as window size in windowed-FFT, mother wavelet functions in wavelet transforms. The success of conventional diagnosis framework largely attributes to the prior knowledge about the analysis objects as well as signal processing algorithms (this can also be observed from the results on two feature sets for shallow methods–SVM and RF, where the performance differs a lot on two feature sets). In this context, an adaptive feature learning framework is beneficial for intelligent fault diagnosis. Note, such hyperparameter choice is involved in deep learning models such as CNN as well. But as long as appropriate model capacity is used for them based on rules of thumb in the community, they tend to be less sensitive to hyperparameter choice and have better accuracy compared to the traditional signal processing approaches.

As the scale of the system increases, more types of measurements need to be processed. In this context, the manual feature extraction approaches are with high cost and not general. For instance, the wind speed and acceleration are both collected in the wind turbine data set. The acceleration is well researched and the handcrafted features (e.g., features extracted from FFT and wavelet packet analysis) are extremely useful for diagnosing the related faults. However, such feature extraction methods mostly do not work on the wind speed. Furthermore, the individual wind speed is probably not helpful for condition classification, while the dependency between the wind speed and other measurements (spatial features) is beneficial for fault diagnosis if they are properly represented. The case study on wind turbine data set (Section 4.1) shows that the spatial features play an important role in the classifier. Also, the proposed framework is open to the features that are well known and shown be useful in classification (as shown in Section 4.2, where 29 statistical features are included).

The two cases discussed in this work are essential for fault diagnosis that there are not adequate training data for every operating condition and fault severity. And this needs the fault diagnosis algorithm can learn the general classifiable features via one or several operating conditions or fault severities, and then it can be used to diagnose untrained or unseen situations.

Table 12
Performance of ST-CNN and comparisons on bearing data set.

Method	Num of variables	Num of conds.	Training samples	Testing[a] samples	20ptAccuracy
SVM-1[b]	1	4	800	800/ 800	100.0/ 64.8[d]
RF-1	1	4	800	800/ 800	100.0/ 67.7
SVM-2[c]	1	4	800	800/ 800	100.0/ 72.2
RF-2	1	4	800	800/ 800	100.0/ 75.7
1D CNN	1	4	16102	2842/ 2842	100.0/ 73.6
ST-CNN	1	4	16102	2842/ 2842	100.0/ 90.6
ST-CNN w TF	1	4	16102	2842/ 2842	99.9/ 92.4

[a] The testing samples are from fault severity 1 and fault severities 2 & 3 respectively.
[b] using feature set 1 in Table 11.
[c] using feature set 2 in Table 11.
[d] The accuracies are based on the two sets of testing samples accordingly.

Table 13
Computation time for 1D CNN and ST-CNN with wind turbine data set 'WT8-12'.

Method	Time (sec/epoch)	Memory (MB)	Trainable parameters
1D CNN	48.8	7,755	592,236
ST-CNN	22.8	851	223,596

Although 1D-CNN shows better accuracy to diagnose seen OCs in some cases, evident performance drop is often accompanied when diagnosing the unseen conditions. The stronger generalization capacity of the proposed framework is verified on both scenarios.

In both cases, the features learnt by STPN are still with high dimensions, which increase the requirement of data and the computational cost. Considering the data compression ability of STPN, further analysis is being carried out with a more adaptive algorithm to reduce the dimensions of the features while preserving the classification ability. Also, the interpretation of the trained ST-CNN model needs more input to understand why the case succeeds or fails and which variable is essential for the fault diagnosis, and this will probably also lead to a more general algorithm for unseen operating conditions or fault severities.

5. Conclusion

To address the important issues that how to efficiently utilize multivariate time-series data from complex mechanical systems for fault diagnosis and improve the classification accuracy in unseen operating conditions and fault severities, this work presents a ST-CNN framework consisting of spatiotemporal feature learning via STPN and condition classification using CNNs. With the wind turbine and bearing data sets, the proposed approach outperforms the shallow methods (SVM and RF) and pure deep learning model–1D CNN, and its computational complexity is less than 1D CNN. The case study on wind turbine test rig validates that the STPN can properly extract the dependency between variables (namely spatial features) and boost the classifier (CNNs) with the spatial features, especially for diagnosing the unseen operating conditions. The case study on the bearing data set shows that the proposed ST-CNN framework can classify unseen fault severities more accurately than the 1D CNN and shallow methods. Furthermore, it shows that the proposed framework is capable of fusing diverse features to improve the performance. With the presented approach, it is promising that the spatiotemporal features among the complex system can be well represented and more general features are determined by the proposed approach to form an adaptive classifier for diverse operating conditions and different fault severities.

Acknowledgements

The authors would like to express their sincere gratitude to Dr. Wenguang Yang and Adedotun Akintayo for their thoughtful comments on the proposed framework. This work was supported by National Natural Science Foundation of China (Grant No. 11572167), and was partially supported by AFOSR YIP Grant (FA9550-17-1-0220) and National Science Foundation under Grant No. CNS-1464279.

References

[1] W. Caesarendra, B. Kosasih, A.K. Tieu, H. Zhu, C.A. Moodie, Q. Zhu, Acoustic emission-based condition monitoring methods: review and application for low speed slew bearing, Mech. Syst. Signal Process. 72 (2016) 134–159.
[2] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, R.E. Vásquez, Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals, Mech. Syst. Signal Process. 76 (2016) 283–293.
[3] C.M. Vicuña, C. Höweler, A method for reduction of acoustic emission (ae) data with application in machine failure detection and diagnosis, Mech. Syst. Signal Process. 97 (2017) 44–58.
[4] S. Lu, Q. He, J. Zhao, Bearing fault diagnosis of a permanent magnet synchronous motor via a fast and online order analysis method in an embedded system, Mech. Syst. Signal Process. 113 (2018) 36–49.
[5] M.E.K. Oumaamar, Y. Maouche, M. Boucherma, A. Khezzar, Static air-gap eccentricity fault diagnosis using rotor slot harmonics in line neutral voltage of three-phase squirrel cage induction motor, Mech. Syst. Signal Process. 84 (2017) 584–597.
[6] F. Gu, T. Wang, A. Alwodai, X. Tian, Y. Shao, A. Ball, A new method of accurate broken rotor bar diagnosis based on modulation signal bispectrum analysis of motor current signals, Mech. Syst. Signal Process. 50 (2015) 400–413.
[7] M. Abd-el Malek, A.K. Abdelsalam, O.E. Hassan, Induction motor broken rotor bar fault location detection through envelope analysis of start-up current using hilbert transform, Mech. Syst. Signal Process. 93 (2017) 332–350.
[8] V. Ghorbanian, J. Faiz, A survey on time and frequency characteristics of induction motors with broken rotor bars in line-start and inverter-fed modes, Mech. Syst. Signal Process. 54 (2015) 427–456.
[9] V. Choqueuse, M. Benbouzid, et al, Induction machine faults detection using stator current parametric spectral estimation, Mech. Syst. Signal Process. 52 (2015) 447–464.
[10] Y. Lu, F. Wang, M. Jia, Y. Qi, Centrifugal compressor fault diagnosis based on qualitative simulation and thermal parameters, Mech. Syst. Signal Process. 81 (2016) 259–273.
[11] T. Touret, C. Changenet, F. Ville, M. Lalmi, S. Becquerelle, On the use of temperature for online condition monitoring of geared systems–a review, Mech. Syst. Signal Process. 101 (2018) 197–210.
[12] S. Aouabdi, M. Taibi, S. Bouras, N. Boutasseta, Using multi-scale entropy and principal component analysis to monitor gears degradation via the motor current signature analysis, Mech. Syst. Signal Process. 90 (2017) 298–316.

[13] R. Zhang, F. Gu, H. Mansaf, T. Wang, A.D. Ball, Gear wear monitoring by modulation signal bispectrum based on motor current signal analysis, Mech. Syst. Signal Process. 94 (2017) 202–213.
[14] L. Jing, T. Wang, M. Zhao, P. Wang, An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox, Sensors 17 (2017) 414.
[15] W. Qiao, D. Lu, A survey on wind turbine condition monitoring and fault diagnosis part i: Components and subsystems, IEEE Trans. Industr. Electron. 62 (2015) 6536–6545.
[16] A. Kusiak, W. Li, The prediction and diagnosis of wind turbine faults, Renewable Energy 36 (2011) 16–23.
[17] W.A. Smith, R.B. Randall, Rolling element bearing diagnostics using the case western reserve university data: A benchmark study, Mech. Syst. Signal Process. 64 (2015) 100–131.
[18] Z. Shen, X. Chen, X. Zhang, Z. He, A novel intelligent gear fault diagnosis model based on emd and multi-class tsvm, Measurement 45 (2012) 30–40.
[19] M. Ma, X. Chen, X. Zhang, B. Ding, S. Wang, Locally linear embedding on grassmann manifold for performance degradation assessment of bearings, IEEE Trans. Reliab. 66 (2017) 467–477.
[20] Y. Guo, J. Na, B. Li, R.-F. Fung, Envelope extraction based dimension reduction for independent component analysis in fault diagnosis of rolling element bearing, J. Sound Vib. 333 (2014) 2983–2994.
[21] L. Han, C.W. Li, S.L. Guo, X.W. Su, Feature extraction method of bearing ae signal based on improved fast-ica and wavelet packet energy, Mech. Syst. Signal Process. 62 (2015) 91–99.
[22] Y. Wang, G. Xu, L. Liang, K. Jiang, Detection of weak transient signals based on wavelet packet transform and manifold learning for rolling element bearing fault diagnosis, Mech. Syst. Signal Process. 54 (2015) 259–276.
[23] F. Jia, Y. Lei, L. Guo, J. Lin, S. Xing, A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines, Neurocomputing 272 (2018) 619–628.
[24] W. Zhang, G. Peng, C. Li, Y. Chen, Z. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, Sensors 17 (2017) 425.
[25] H. Shao, H. Jiang, H. Zhang, W. Duan, T. Liang, S. Wu, Rolling bearing fault feature learning using improved convolutional deep belief network with compressed sensing, Mech. Syst. Signal Process. 100 (2018) 743–765.
[26] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, Mech. Syst. Signal Process. 100 (2018) 439–453.
[27] M. Xia, T. Li, L. Xu, L. Liu, C.W. de Silva, Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks, IEEE/ASME Trans. Mechatron. (2017).
[28] H. Shao, H. Jiang, Y. Lin, X. Li, A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders, Mech. Syst. Signal Process. 102 (2018) 278–297.
[29] L. Jing, M. Zhao, P. Li, X. Xu, A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox, Measurement 111 (2017) 1–10.
[30] C. Liu, Y. Gong, S. Laflamme, B. Phares, S. Sarkar, Bridge damage detection using spatiotemporal patterns extracted from dense sensor network, Meas. Sci. Technol. 28 (2017) 014011.
[31] Z. Jiang, C. Liu, A. Akintayo, G.P. Henze, S. Sarkar, Energy prediction using spatiotemporal pattern networks, Appl. Energy 206 (2017) 1022–1039.
[32] C. Rao, A. Ray, S. Sarkar, M. Yasar, Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns, SIViP 3 (2009) 101–114.
[33] S. Sarkar, K.G. Lore, S. Sarkar, Early detection of combustion instability by neural-symbolic analysis on hi-speed video, in: Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (CoCo@ NIPS 2015), Montreal, Canada, 2015.
[34] C. Liu, A. Akintayo, Z. Jiang, G.P. Henze, S. Sarkar, Multivariate exploration of non-intrusive load monitoring via spatiotemporal pattern network, Appl. Energy 211 (2018) 1106–1122.
[35] S. Sarkar, A. Srivastav, A composite discretization scheme for symbolic identification of complex systems, Signal Processing 125 (2016) 156–170.
[36] S. Sarkar, S. Sarkar, K. Mukherjee, A. Ray, A. Srivastav, Multi-sensor data interpretation and semantic fusion for fault detection in aircraft gas turbine engines, Proc. I Mech. E Part G: J. Aerospace Eng. 227 (December 2013) 1988–2001.
[37] C. Liu, S. Ghosal, Z. Jiang, S. Sarkar, An unsupervised anomaly detection approach using energy-based spatiotemporal graphical modeling, Cyber-Phys. Syst. (2017) 1–37.
[38] W. Yang, C. Liu, D. Jiang, An unsupervised spatiotemporal graphical modeling approach for wind turbine condition monitoring, Renewable Energy 127 (2018) 230–241.
[39] C. Liu, S. Ghosal, Z. Jiang, S. Sarkar, An unsupervised spatiotemporal graphical modeling approach to anomaly detection in distributed CPS, in: Proceedings of the International Conference of Cyber-Physical Systems, (Vienna, Austria), 2015.
[40] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.
[41] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3431–3440.
[42] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.
[43] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.
[44] X. Zhang, Y. Liang, J. Zhou, et al, A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized svm, Measurement 69 (2015) 164–179.
[45] C. Liu, D. Jiang, W. Yang, Global geometric similarity scheme for feature selection in fault diagnosis, Expert Syst. Appl. 41 (2014) 3585–3595.
[46] T. Han, D. Jiang, Q. Zhao, L. Wang, K. Yin, Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery, Trans. Inst. Measure. Control (2017) (0142331217708242).
[47] E. Tsironi, P. Barros, C. Weber, S. Wermter, An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition, Neurocomputing (2017).
[48] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5353–5360.
[49] L.V.D. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learning Res. 9 (2008) 2579–2605.
[50] R.R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-cam: Why did you say that?, arXiv preprint arXiv:1611.07450, 2016.
[51] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, IEEE, 2016, pp. 2921–2929.
[52] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, vol. 1, MIT press Cambridge, 2016.
[53] B.D. Center, Case western reserve university bearing data, 2013.