Received October 20, 2019, accepted October 30, 2019, date of publication November 1, 2019, date of current version November 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950985

A Review on Deep Learning Applications in Prognostics and Health Management

LIANGWEI ZHANG[1,2], JING LIN[2], (Member, IEEE), BIN LIU[3], (Member, IEEE), ZHICONG ZHANG[1], XIAOHUI YAN[1], AND MUHENG WEI[4]

[1]Department of Industrial Engineering, Dongguan University of Technology, Dongguan 523808, China
[2]Division of Operation and Maintenance Engineering, Luleå University of Technology, 971 87 Luleå, Sweden
[3]Department of Management Science, University of Strathclyde, Glasgow G1 1XQ, U.K.
[4]Oceanic Intelligent Technology Innovation Center, CSSC Systems Engineering Research Institute, Beijing 100073, China

Corresponding author: Bin Liu (b.liu@strath.ac.uk)

This work was supported in part by the National Natural Science Foundation of China under Grant 71801045, Grant 61703102, and Grant 61633001, in part by the Youth Innovative Talent Project from the Department of Education of Guangdong Province, China, under Grant 2017KQNCX191, in part by research start-up funds of DGUT under Grant GC300502-46, in part by the Luleå Railway Research Centre (Järnvägstekniskt Centrum), Sweden, and in part by the Swedish Transport Administration (Trafikverket).

ABSTRACT Deep learning has attracted intense interest in Prognostics and Health Management (PHM), because of its enormous representing power, automated feature learning capability and best-in-class performance in solving complex problems. This paper surveys recent advancements in PHM methodologies using deep learning with the aim of identifying research gaps and suggesting further improvements. After a brief introduction to several deep learning models, we review and analyze applications of fault detection, diagnosis and prognosis using deep learning. The survey validates the universal applicability of deep learning to various types of input in PHM, including vibration, imagery, time-series and structured data. It also reveals that deep learning provides a one-fits-all framework for the primary PHM subfields: fault detection uses either reconstruction error or stacks a binary classifier on top of the network to detect anomalies; fault diagnosis typically adds a soft-max layer to perform multi-class classification; prognosis adds a continuous regression layer to predict remaining useful life. The general framework suggests the possibility of transfer learning across PHM applications. The survey reveals some common properties and identifies the research gaps in each PHM subfield. It concludes by summarizing some major challenges and potential opportunities in the domain.

INDEX TERMS Condition-based maintenance, deep learning, fault detection, fault diagnosis, prognosis.

I. INTRODUCTION

Prognostics and Health Management (PHM) has emerged as a critical approach to achieving a competitive edge in many industries because of its potential for reliability, safety and cost reduction. PHM uses measurements, models and software to perform incipient fault detection, condition assessment and failure progression prediction [1], [2]. It provides users with the ability to perceive the health state of a part, asset, subsystem or system [3]. As shown in Fig. 1, a holistic PHM framework typically incorporates data collection, data manipulation, fault detection, fault diagnosis, prognosis and decision support in sequential order [2], [4], [5]. Of these, fault detection, diagnosis and prognosis are the most researched [6], [7]. Therefore, in this paper, we restrict our review to these three topics.

PHM methods can be roughly classified as either physics model-based or data-driven [2], [8], [9]. The former requires knowing the first principles of the item under investigation, such as material properties, structural characteristics and failure mechanisms [10], [11]. While highly accurate when applied to the component level, the method may not perform well in modern complex systems because intra-system interactions often occur in very complicated ways and cannot be easily captured by physical models [9]. Data-driven methods attempt to acquire hidden knowledge from empirical data, to infer current health states of the item of interest and to predict its Remaining Useful Life (RUL) [7], [12]. Data-driven methods can be further divided into supervised and unsupervised approaches, depending on whether the raw data

The associate editor coordinating the review of this manuscript and approving it for publication was Faisal Khan.

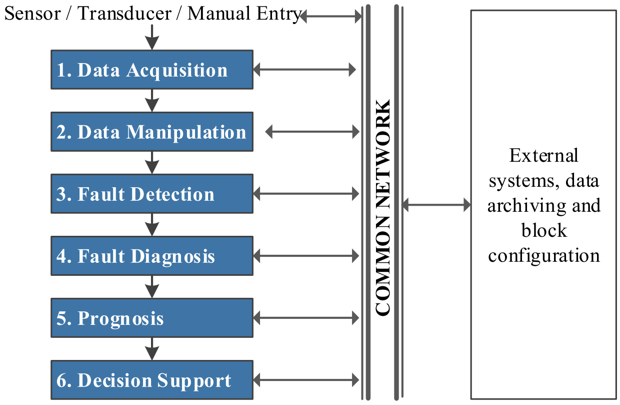


FIGURE 1. Holistic PHM framework, adapted from Open System Architecture for CBM (OSA-CBM) and ISO 13374-2:2007 [2], [4], [5].

are labeled or not. With the data deluge in industry and ever-increasing computing power, data-driven methods are finding more opportunities in PHM applications [13], for example, in [14] and [15].

Advances in sensor technology and Information and Communication Technologies (ICT) have led to the creation of "Industrial Big Data." These data tend to be multi-modal, unstructured, decentralized, heterogeneous, fast-flowing and highly nonlinear [16], posing significant challenges to traditional data-driven methods in PHM applications. For example, in two studies, Unmanned Aerial Vehicles (UAV) were used to carry out regular inspection of railway tracks [17], [18]. Images and videos taken by the drones can be analyzed to detect potential track defects, such as squats, poor-quality insulated joints, structural damage and so on. However, traditional methods of analysis rely on domain expertise to extract useful features like edges, lines and textures, which can then be fed to other learning algorithms [17], [18]. These hand-crafted features may be subjective, implying low efficiency and high labor cost. Traditional methods also require a large number of labeled samples for training. It is hard to meet this requirement in many real-world applications where experiments are costly or even not allowed. In another study, researchers showed how bearing RUL could be predicted using vibration data [19]. Thanks to advancements in artificial intelligence, deep learning provides a way to meet the challenges of Big Data.

Deep learning comes from research into Artificial Neural Networks (ANNs), where "deep" contrasts itself with conventional shallow neural networks in terms of the depth of the network architecture [20]. The use of deep architecture, extensible hidden units and nonlinear activation functions gives a deep neural network with an ability to model complex data, such as acoustic data, natural language and images; see the universal approximation theorem [21]. One, maybe the most attractive aspect of deep learning is that it can automate feature engineering, the learning of internal representation and the creation of feature vectors of the raw data without human intervention [20], thus alleviating the need for domain expertise and hardcore feature extraction. The learned features are typically stacked layer-wisely, with high-level ones more abstract than lower ones; the high-level representations can detect, classify and predict patterns in the input. In addition, the incorporation of feature learning into a deep neural network allows parameters in a feature engineering module and a pattern recognition module to be jointly trained, leading to better performance [22]. It also enables end-to-end learning, making deep learning models generic in PHM, i.e., not restricted to a specific piece of equipment or a particular application. In other words, deep learning models can be adapted to new problems relatively easily. Many researchers have reused pretrained networks to solve their problems in PHM with an effortless modification in the architecture and a finetuning process; see Section III for concrete examples. This is generally called transfer learning, i.e., transferring the knowledge learned from a source problem to a similar but different target problem. The use of transfer learning can greatly reduce the need for labelled samples in the target problem. All the above properties of deep learning make its performance best-in-class in many complex problems.

Many researchers have applied deep learning technologies to PHM applications. Some focus on a subfield of PHM, e.g., fault diagnosis or prognosis [23], [24]; others focus on applications to a specific item, e.g., bearing or electronic system [25]–[27], while still others survey PHM applications from the point of view of various deep learning architectures [22], [28]. However, none provides a comprehensive survey of the full coverage of the PHM domain from an application perspective. Besides, the major problems in the field are: studies to various PHM subfields are somewhat independent from each other, leading to a lack of sharing of data, models and knowledge; existing researches share many commonalities, yet scholars are still reinventing the wheels; there are no guidelines on the design or selection of a "good" deep learning model for different applications; a unified evaluation system to different methods is still missing. Though this paper is not intended to solve all the above problems, we do hope it may induce others to come forward with valuable contributions.

In this paper, we survey recent advancements of PHM methodologies using deep learning, with a focus on their applications in fault detection, diagnosis and prognosis. In response to the emerging challenges, as well as the opportunities, we identify research gaps that when filled may lead to the improvement of PHM in both theory and practice. The major contributions of this paper are the following:

1) It presents a comprehensive review of deep learning applications in fault detection, diagnosis and prognosis. To enable systematical analysis, the applications are categorized according to their type of input: vibration (incl. acoustic), imagery, structured and time-series data. With this design, the paper may serve as a reference for researchers looking for studies related to their work. It also validates the universal applicability of deep learning to various types of data in PHM.

2) The review leads us to conclude that deep learning provides a one-fits-all framework for fault detection, diagnosis and prognosis. This makes transfer learning possible, allowing a pretrained network to be reused across different PHM subfields. In other words, given the same type of input, we can adapt a deep learning model from one PHM application to another with minimal effort. This could significantly reduce the number of required labeled samples. Notably, the use of transfer learning across different PHM subfields are rarely reported in the literature.
3) For each PHM subfield, we find some common properties, such as the design of an appropriate network architecture, the selection of loss functions and evaluation metrics. These common properties can be a guideline for future studies. In addition, we mention some benchmarking datasets and compare existing research referring to these datasets, with the hope of recognizing the best practices. We also identify the research gaps within each PHM subfield.
4) The paper summarizes five challenges and three opportunities based on the review. The five challenges are: the artistic use of deep learning technologies; poor generalization to real-world applications; the ''concept drift'' problem; the timeliness concern; the creation of actionable tasks. The three opportunities are: transfer learning, data augmentation and end-to-end learning.

The remainder of this paper is organized as follows. Section II briefly introduces four commonly used types of deep learning architecture and their variants. Section III reviews research adopting deep learning in fault detection, fault diagnosis and prognosis. Section IV presents current challenges of and opportunities for using deep learning in PHM. Finally, Section V concludes the work.

II. DEEP LEARNING MODELS

With the theoretical development of deep learning, various network architectures have been proposed for different domains, ranging from speech recognition to computer vision, natural language processing, learning from structured data and so on [20]. Each has its own specialties and frailties in dealing with different types of data. In PHM, the data available for training are widely variable, including, for instance, thermal infrared images, vibration signals, stator current and multiple sensor fusion. In this section, we briefly touch on four widely used architectures in the PHM domain: auto-encoder (AE), restricted Boltzmann machine (RBM), convolutional neural network (CNN), recurrent neural network (RNN) and their variants.

A. AUTO-ENCODER AND ITS VARIANTS

An auto-encoder has a feed-forward network architecture which can learn feature representations of input data without supervision. It consists of two components, i.e., an encoder and a decoder, as shown in Fig. 2. The encoder compresses input data to hidden layers with a smaller number of neurons, from which the decoder tries to reconstruct the input [29]. Training an auto-encoder requires minimizing the average reconstruction loss, typically the squared error function over a given training set.

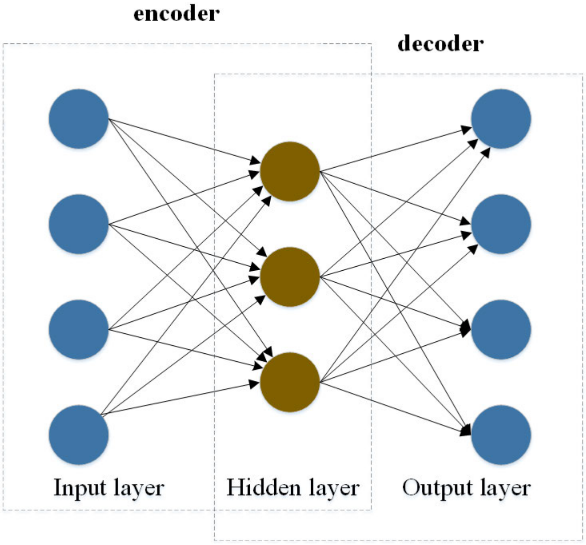


FIGURE 2. Architecture of an auto-encoder, containing two components: an encoder and a decoder.

The intuition behind an auto-encoder is as follows: if the decoder obtains a good reconstruction of the input, the neurons in the hidden layers must preserve the vast majority information of the original data. The shrinkage in the size of hidden neurons forces the network to learn representative features of the input. The use of nonlinear activation functions, such as relu, tanh and sigmoid, enables the network to learn complex and useful feature representations, and the deep depth of the architecture gives the network the chance to learn hierarchical and more abstract features. It is worth noting that greedy layer-wise pretraining can be performed on auto-encoders to learn hierarchical feature representations of input, as detailed in the next subsection.

Variants of auto-encoders include the following:

1) Sparse Auto-Encoder (SAE): By imposing sparsity constraints on the hidden neurons, an auto-encoder can learn sparse feature representations of the input [30]. More specifically, it adds a sparsity cost term to the loss function; the sparsity cost term measures the KL-divergence between a target activation level and the average activation value of all hidden units. In this way, the activation of hidden units is suppressed, leading to a sparse representation.
2) Denoising Auto-Encoder (DAE): DAE takes in stochastically corrupted data and tries to denoise a clean version from the corrupted one. It can make the reconstruction more robust and prevent the network from learning the identity transformation [31]. The most common way to corrupt the data is to introduce dropout noise or binary masking noise; this randomly sets a fraction of elements in the input to zeros. Isotropic Gaussian noise is also used occasionally.

3) Contractive Auto-Encoder (CAE): CAE encourages invariance or robustness for small variations of input by adding a penalty term to the loss function; this penalty term, the Frobenius norm of the Jacobian of the nonlinear mapping 32], can learn more robust representations of the input.

B. RESTRICTED BOLTSMANN MACHINE AND ITS VARIANTS

The architecture of a restricted Boltzmann machine is a bipartite graph, in which two groups (or layers) of nodes – visible units and hidden units – are fully interconnected, with no intra-layer connection in the graph. Visible units take in input data, and hidden units are feature representations of them, as shown in Fig. 3.

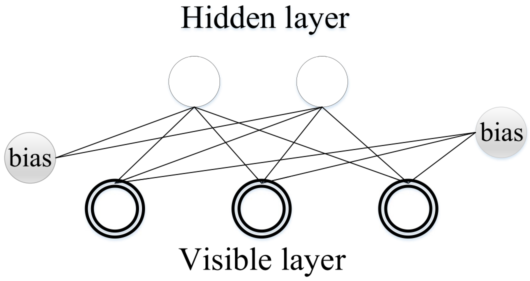


FIGURE 3. Architecture of a restricted Boltzmann machine, which is essentially an undirected probabilistic graphical model.

RBMs are undirected Probabilistic Graphical Models (PGMs). All the visible/hidden units are conditionally independent on the hidden/visible units [33]. By iteratively updating the network connection weights and bias units using an algorithm called contrastive divergence, the log likelihood of a given dataset with respect to the network parameters can be maximized. This leads to a useful feature representation of the input in the hidden layer; from the hidden layer one can reconstruct the input approximately, in much the same way as an auto-encoder. Two RBM variants are the following:

1) Deep Belief Network (DBN): A DBN can be constructed by stacking multiple RBMs on top of each other. The output of the i-th hidden layer serves as visible units of the $(i+1)$-th hidden layer. Except for the undirected connections between the two layers farthest from the visible layer, all connections among all other layers are directed [34]. A DBN is typically trained using an unsupervised, greedy, layer-wise pretraining, followed by a back propagation finetuning [35]. Layer-wise pretraining provides a good initialization to the network parameters, while finetuning adjusts the parameters to fit the target more accurately.
2) Deep Boltzmann Machine (DBM): By extending a simple RBM's single hidden layer to multiple hidden layers, we obtain a deep Boltzmann machine. Unlike DBN, which is a mixed directed PGM, a DBM is fully undirected with cross-layer connections but no within-layer connections [36]. This requires the network to be trained jointly, calling for a sophisticated and computationally expensive training algorithm [37]. DBMs can learn complex structures and construct hierarchical feature representations of input data.

C. CONVOLUTIONAL NEURAL NETWORK

CNN was originally proposed by LeCun in a handwritten digit recognition task [38]. Since then, researchers have repeatedly proven its success in various applications, including computer vision, natural language processing and speech recognition. CNN has a feedforward architecture, consisting of two fundamental operators: convolution and subsampling (also known as pooling), as shown in Fig. 4. The convolution operator extracts local features from the input using different filters (also known as kernels). One unique characteristic of CNN is that the filters can be learned automatically instead of being handcrafted. The subsampling operator extracts the most significant local features from the output of a convolutional layer. It may reduce the dimensionality of an intermediate layer, consequently avoiding overfitting. Another merit of the pooling operation is the translation and rotation invariance property the network can achieve [39].

The stack of multiple convolutional layers and pooling layers allows a CNN to learn hierarchical feature representations of the input. The deeper the layer, the more abstract the feature representation that can be learned. Typically, the final layers of a CNN are constructed with fully connected layers, followed by an output layer associated with the target prediction. The training of a CNN can use the famous Back Propagation (BP) algorithm. CNNs spatially (or temporally) exploit local correlations by enforcing a local connectivity pattern, i.e., sparse connectivity, between neurons of contiguous layers. Sparse connectivity mimics the behavior of the local receptive field in the human brain, a concept from neuroscience. Sparse connectivity, together with the weight sharing mechanism, also reduces the number of network parameters significantly, making the network less prone to overfitting.

The network architecture shown in Fig. 4 is a 2D CNN. It uses 2D filters and the convolution operation is conducted on both the lateral and longitudinal dimension of the input. One variant of CNN is the 1D CNN, which employs 1D filters to convolve along single dimension of its inputs. Though applicable for 2D inputs, 1D CNNs are mainly tailored for 1D inputs, such as acoustic, electrocardiogram signal, etc. In contrast to 2D CNNs which relies on massive matrix operations, 1D CNNs adopt simple array operations. This makes 1D CNNs much less computationally demanding.

D. RECURRENT NEURAL NETWORK AND ITS VARIANTS

To encode temporal information in sequential data, a recurrent neural network defines unique topological connections between neurons. The hidden state at time step t can receive a signal from the input at current time t, as well as from the output of the hidden state at previous time t-1, allowing the

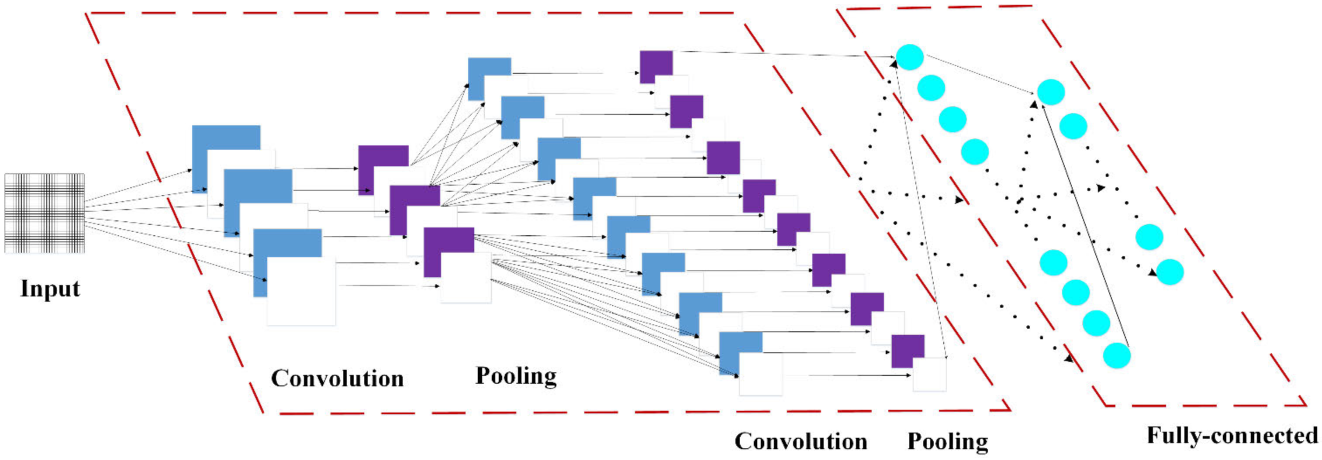


FIGURE 4. Architecture of a 2D convolutional neural network, stacked with a fully-connected layer.

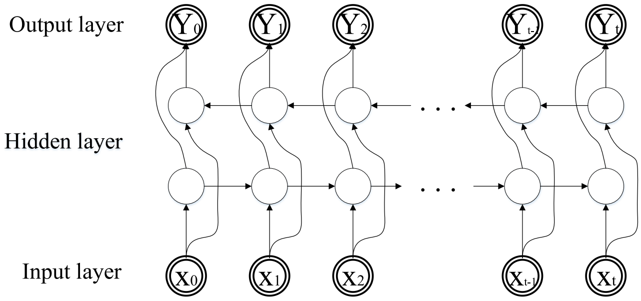


FIGURE 5. Architecture of a *bi-directional recurrent neural network.*

memory of previous inputs to be maintained in the network, as shown in Fig. 5. An RNN takes in sequential data, propagates calculations through hidden states step by step and then yields an output [40]. Neurons in the output layer of an RNN may have varying sizes, depending on the specific application.

Unlike traditional neural networks, which use different parameters at each layer, an RNN shares the same set of parameters across all steps. This greatly reduces the total number of parameters and forces the network to learn important features from the sequences. Like many other deep neural networks, the RNN keeps feature representations of input sequences in hidden layers. A stacked RNN is constructed by stacking multiple hidden layers on top of each other.

To maintain long-term memory (i.e., long temporal dependencies in sequential data), vanilla RNNs need to have deep recurrent architecture. However, the training of a vanilla RNN may suffer from the gradient vanishing problem. To solve this, two variants of RNN were proposed: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) network. The key of LSTM and GRU lies in the introduction of a gating mechanism, which allows important features in the input stream for a long series of steps to be maintained as it is instead of being overwritten invariably. GRU is a simplified version of LSTM, but it is comparable to LSTM in terms of generalization capability.

Another RNN variant, a Bidirectional RNN (BRNN), attempts to exploit temporal information in sequential data. BRNNs encode temporal information not only in the forward direction, but also in the backward direction [43]. Consequently, the hidden states in a BRNN depend on both past and future states. By replacing the basic hidden units in a BRNN with LSTM units or GRU units, we can obtain bidirectional LSTM or bidirectional GRU networks. The increased complexity of these variants allows them to be more flexible and powerful than a simple RNN.

III. APPLICATIONS OF DEEP LEARNING IN PHM

Deep learning is increasingly popular in PHM applications because of its powerful representing capability and its universal applicability to various types of data. However, the difficulties of needle-threading all the steps in the PHM framework using deep learning, and outputting actionable and reasonable recommendations are still formidable. In the literature, most related work studies the PHM subfields individually, mainly fault detection, fault diagnosis or prognosis; very few focuses on decision making. Accordingly, we organize our review into the following three subsections and categorize existing work according to the type of data input. Fig. 6 gives an overview of this section.

A. FAULT DETECTION

Human beings are capable of sensing their own illness, even if they do not know the exact nature of the illness. Machines should be endowed with a similar self-aware intelligence. Fault detection, also called anomaly detection, aims to detect instances which deviate so much from others that they are suspected of being generated by different mechanisms [44]. Fault detection can be simplified to a binary classification task, i.e., to classify whether the item of interest is working well or if something has gone wrong [5]. Depending on the availability of positive (faulty) samples, fault detection applications using deep learning can be grouped into two categories: supervised and unsupervised.

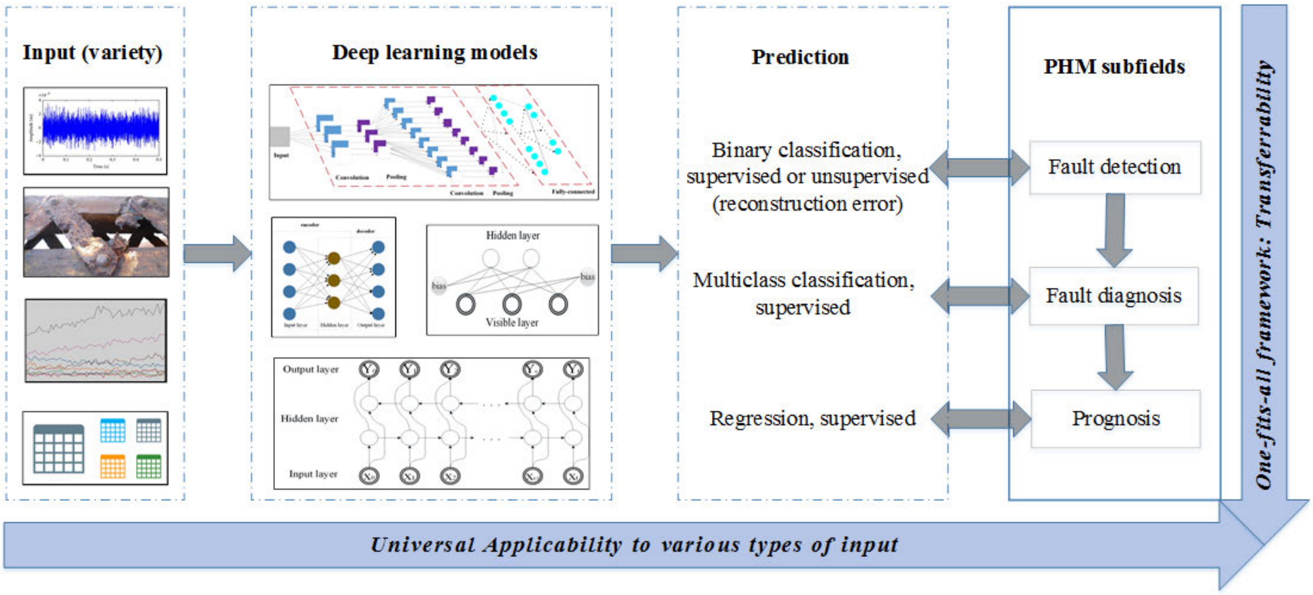

Input (variety)
Deep learning models
Prediction
PHM subfields
Binary classification, supervised or unsupervised (reconstruction error)
Multiclass classification, supervised
Regression, supervised
Fault detection
Fault diagnosis
Prognosis
Hidden layer
Visible layer
bias
Output layer
Hidden layer
Input layer
One-fits-all framework: Transferrability
Universal Applicability to various types of input

FIGURE 6. Deep learning provides a one-fits-all framework to all the major PHM subfields, including fault detection, fault diagnosis and prognosis. It also has universal applicability to various types of input in the PHM domain, mainly vibration, imagery, time-series and structured data.

1) SUPERVISED LEARNING

When there are enough faulty data, may not limit to a certain faulty type, a classifier can be constructed to discriminate faulty from normal states. The classifier tries to learn a function, mapping from sensor measurements to their state labels with the aim of separating the two classes. But in most real-world scenarios, data available for training have a skewed class distribution, also known as imbalanced classes, the majority of which are negative (normal samples) and a minority positive. In such circumstances, techniques like data augmentation, oversampling or under-sampling and stratified cross validation should be integrated into the learning process to improve the generalization capability [45]. In general, these techniques can fit any machine learning tasks facing the class imbalance problem; hence, we do not provide further details of this. Instead, we focus on deep learning related topics.

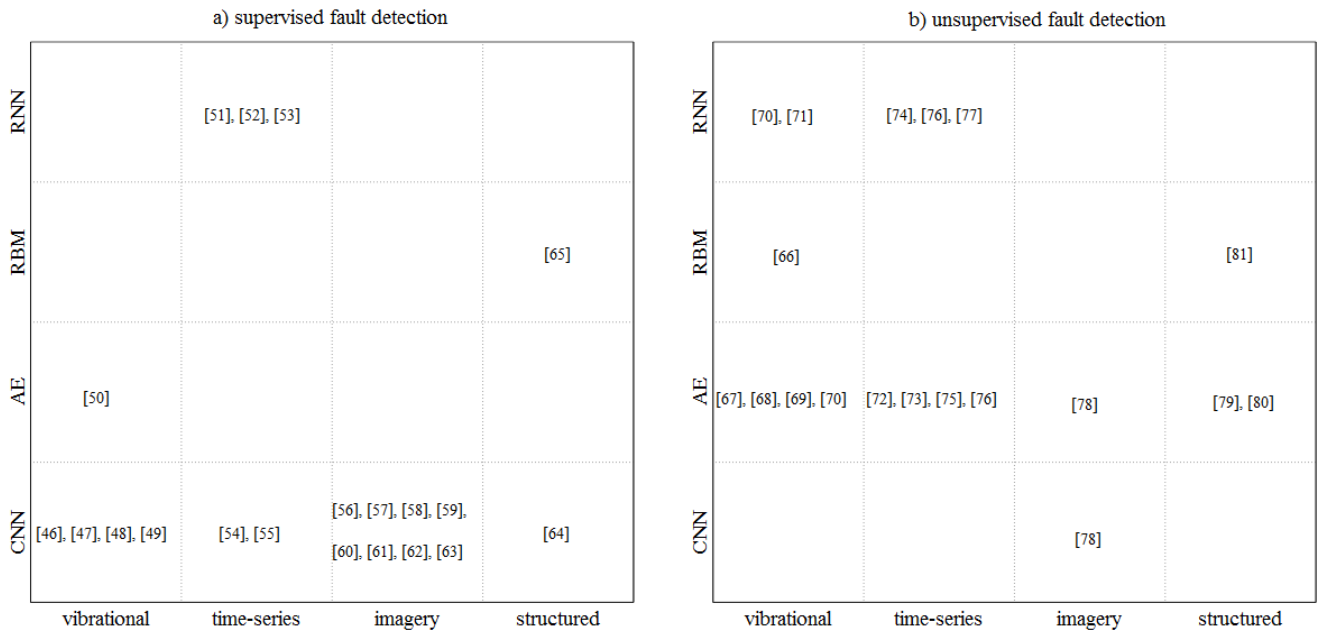
The selection of a particular deep learning model in fault detection is dependent on the type of data available and the application domain. To structure our analysis, we divide the literature into four categories according to the type of input data: vibration (incl. acoustic data), time-series, imagery (incl. video frames) and structured data.

Vibration (incl. acoustic data) data play a major role in detecting and diagnosing faults in rotating or reciprocating equipment. In fact, this is the most researched subject in PHM. The prevailing deep learning model for vibration data is CNN, as shown in Fig.7. Janssens *et al.* built a classifier to detect bearing faults from the vibration signal [46]. In their use of CNN, the internal representation of the vibration signal was captured by two perpendicularly mounted accelerometers. They then used random forests to classify the learned high-level features. The main point is to use CNN's capability to exploit the spatial structure in data to capture the covariance of the frequency decomposition of the data. By doing so, the model can differentiate between complex bearing conditions by learning the patterns of changes in the joint accelerometer signal.

Abdeljaber *et al.* used 1D CNN to detect structural damage using vibration signals [47]. They fed raw vibration signals directly to the 1D CNN model and outputted a binary label (damaged/undamaged), forming end-to-end learning. This method allows feature engineering and classification to be jointly trained, leading to better accuracy.

In their study, Bach-Andersen *et al.* first extracted frequency domain features from the vibration data of a wind turbine drivetrain using traditional signal analysis techniques [48]. Then they down-sampled the frequency spectrum to some predetermined dimensions and fed the spectrum data to their CNN model. Though superior accuracy was reported, the hand-crafted features may not be better than the features learned from an end-to-end architecture. Similar work appears in [49].

The use of other deep learning models includes work by Luo *et al.* [50]. These researchers built an architecture comprised of stacked SAE and fully-connected layers to distinguish impulse responses from non-impulse responses in the vibration data of machine tools. They pretrained their SAE layers with vibration data frames in an unsupervised fashion and finetuned the whole network using back propagation under supervision. They compared time domain, frequency domain, time-frequency domain and SAE features. SAE features were found to be more accurate and stable than traditional signal-based features in classifying the two types of



a) supervised fault detection

	vibrational	time-series	imagery	structured
RNN		[51], [52], [53]		
RBM				[65]
AE	[50]			
CNN	[46], [47], [48], [49]	[54], [55]	[56], [57], [58], [59], [60], [61], [62], [63]	[64]

b) unsupervised fault detection

	vibrational	time-series	imagery	structured
RNN	[70], [71]	[74], [76], [77]		
RBM	[66]			[81]
AE	[67], [68], [69], [70]	[72], [73], [75], [76]	[78]	[79], [80]
CNN			[78]	

FIGURE 7. Fault detection applications: selection of deep learning models given different types of data input, a) supervised cases, b) unsupervised.

responses. The results prove the value of integrating feature learning into a deep learning model.

Although vibration data are collected in the form of time series, their sampling frequency is typically significantly higher than ordinary time-series data. The source of time-series data is not restricted to one type of sensor; multiple sensors can be fused together. The key point in time-series modeling is to capture temporal information which may reflect the health status of the monitored asset. Unsurprisingly, RNN is the most favorable model to deal with time-series data, as shown in Fig. 7. One of the earliest papers using RNN for fault detection was by Hu *et al.* [51]. They constructed a very simple RNN architecture to model a bi-process combining software fault detection and correction; its prediction accuracy outperformed feedforward ANN. Zhang *et al.* used an LSTM network to capture long-term dependencies in time-series data to detect line trip faults in a power system [52]. Taking current, voltage and active power data as the input, they built three separate LSTMs, the outputs of which were concatenated and further fed to an SVM classifier. Obst built an RNN to learn spatial-temporal correlations between sensors in a distributed wireless sensor network [53]. The residuals between actual sensor readings and the RNN predictions were used to detect sensor faults. It is noteworthy that anomalous patterns may exist in the covariation between multiple sensors, even though each sensor signal behaves normally. Overall, the above research shows the value of data fusion.

CNN is another architecture commonly adapted for time-series data. Guo *et al.* constructed a CNN to detect faulty feeders in zero-sequence current waveform acquired from power distribution systems [54]. Interestingly, they applied continuous wavelet transform to the raw current signal and inputted the obtained time-frequency grey scale images to the CNN. They reported that accurate and robust predictions are possible with the proposed method. Ince *et al.* also dealt with current signals using CNN [55]. They inputted the raw signal to a 1D CNN and stacked it with fully connected layers, similar to the method proposed in [47]. The temporal information in time-series data can be captured by a CNN because of the sliding window mechanism in its convolutional operation, and the size of the kernel may have a great impact on the length of the learnable temporal dependencies.

Imagery data are attracting attention in fault detection applications. As has been repeatedly proven in the field of computer vision, CNN can achieve state-of-the-art performance in classifying imagery data. Gibert et al. proposed using CNN to inspect railway tracks, specifically to detect broken or missing fasteners [56]. They trained the network for two purposes: track inspection and material identification. The multi-task learning setting allowed the knowledge learned in one task to be transferred to another (i.e., transfer learning), forming a mutually beneficial mechanism. Similar work used CNN for railway track inspection [57], road pavement crack detection [58], and concrete crack detection [59].

Video data are a composition of images along the temporal dimension. However, if its architecture is not modified, CNN is not good at encoding temporal information. Most existing research has used CNN to learn patterns from video frames, essentially images, and then model the temporal dependencies, as shown in Fig. 7. Chen and Jahanshahi built a CNN to detect crack patches in video frames of metallic surfaces in a nuclear power plant [60]. Their CNN model does not encode temporal information itself, but the output forms spatiotemporal registration tubelets which can be fed to a Naïve Bayes classifier to detect crack patches. Jha *et al.* combined CNN and the Gaussian process to detect instable combustion behavior from high-speed grey-scale videos of a swirl-stabilized combustor [61]. They used CNN to extract features from video frames and adopted the Gaussian process to model the dynamics in the sequential images. As reported in the paper, the model generalizes better when the Gaussian process is added.

Depending on their complexity, the imagery data may not be readily fed to a CNN, outputting the desired state label. Sometimes the object of interest needs to be located in an image before classification can take place. Chen *et al.* proposed a three-stage architecture to locate and detect defective fasteners in images of a catenary support device [62]. The first stage used the SSD (i.e., Single-Shot multi-box Detector) framework to locate cantilever joints; the second employed the YOLO (i.e., You Only Look Once) framework to locate six different fasteners, and the third used a primitive CNN to detect missing or potentially missing fasteners. In a similar process, Lei and Sui used the Faster R-CNN architecture to locate and detect broken insulators and bird nests in high voltage line images [63]. It is worth noting that they reused the pretrained ResNet-101 network, a very deep network trained for the ImageNet competition, to initialize their detection network. This strategy allows the knowledge learned in one field to be transferred to another (i.e., transfer learning). It shortens the time for training and reduces the number of required labelled samples. We discuss this in Section IV.

Last but not least, structured data constitute a major source of fault detection in industry. In contrast to the abovementioned three types of data, structured data may be multi-sourced, distributed and heterogeneous, requiring considerable effort in data fusion and preprocessing. From an algorithmic perspective, structured data have been heavily approached using conventional machine learning techniques, such as SVM, random forest, and feedforward neural network with shallow architectures. The key is to find good feature representations that can be discriminative in separating positive from negative samples. To this end, Chen *et al.* proposed a CNN-based architecture to learn deep representation of SCADA (i.e., Supervisory Control and Data Acquisition) data to detect icing accretion faults in wind turbines [64]. Their input data included 22 measurements related to wind, energy and temperature, and the output was a high-dimensional embedded feature space that could preserve within-class and between-class information while having high discriminative capability. Mandal *et al.* built a DBN to detect faults in a fast breeder test reactor [65]. They fed 175 thermocouple readings into the DBN and output a binary label, indicating faulty or normal. In short, complex cross-correlations between multiple sensors can be captured using deep architecture and nonlinear transformation.

The selection of deep learning models in fault detection depends on the application domain and the type of data available, but there are some common practices across models. First, in model design, the backbone architecture is typically stacked with a logistic layer as the final layer, implying that cross-entropy loss can be used. Second, in the learning process, regularization techniques such as dropout and weight decay are usually adopted to prevent overfitting, and the amount of regularization is a hyperparameter that needs to be tuned. Third, precision, recall, ROC (Receiver Operating Characteristics) curve, AUC (Area Under the Curve) and F-score are commonly used metrics to evaluate model accuracy. Although many applications have been reported in the literature, the limitations of supervised approaches originate in the difficulty of obtaining faulty data. Most research uses data from laboratory tests, but these data are generally insufficient in the real world [9]. Moreover, the generalization capability of supervised approaches to situations that have not yet happened (‘‘unhappened’’ faults) is poor.

2) UNSUPERVISED LEARNING

When there is a lack of sufficiently labelled data, often the case in reality, fault detection may resort to unsupervised methods. In an unsupervised setting, normal operating conditions are modeled beforehand, and faults are detected as deviations from the normal behavior in a process also known as the one-class classification problem. Intuitively, this problem tries to learn patterns from negative samples, specifically, to find a low-dimensional embedding that can capsulize most informative features, from which the samples can be reconstructed with minimal information loss. If a test sample cannot be well reconstructed from its feature embedding, we are tempted to doubt the normality of its generating mechanism. In contrast to supervised learning where a hard rule, e.g., a separating hyperplane, is used to generate a binary output, unsupervised methods often spit out a continuous score representing the abnormality of a given sample; the larger the value, the more its anomalousness, and vice versa. In practice, a threshold is then needed to assist judgment of the occurrence of faults. However, the process is not easy and very much application-dependent because the goal of fault detection is to minimize the chance of committing both Type I error (false positive, or false alarm) and Type II error (false negative, or missed detection), and the cost of these two errors may vary significantly for different applications. In the following section, we survey the relevant research and organize it according to the data type.

Vibration signals are still the major form of input in unsupervised fault detection applications, although the preference for deep learning architecture has obviously shifted the choice towards AE-based ones, as shown in Fig. 7. Sun *et al.* built a model to detect defective electro-motors from vibration signals [66]. They applied greedy layer-wise training on the cepstrograms of vibration clips of normal conditions to learn several RBMs, and then stacked them to form an encoder-decoder-like DBN architecture. Testing samples were fed into the learned DBN, and reconstruction errors between input and output were the criteria to judge their extent of abnormality. Oh and Yun used an AE to detect faults in surface mounting devices using machine sound [67]. They trained the AE with normal data to retain as much information as possible in the bottleneck layer. The residual error between a testing sample and the output of the AE, given the testing sample as the input, was the anomalous score. To shorten the time for training and putting the anomaly detector into production, Park and Yun proposed replacing the basic fully-connected layer in an AE with a stacked LSTM layer in the same application context as above [68]. Both the number of parameters in the network and the training time were significantly cut down at the sacrifice of tolerable accuracy reduction. Using a similar idea but in a different field (i.e., to detect electric motor faults), Principi *et al.* compared the performance of three AEs with different building blocks: a fully-connected layer (what they called MLP), a convolutional layer and an LSTM layer [69]. With meticulous tuning of the hyperparameter, they found that AEs with a fully-connected layer or an LSTM layer outperformed, in terms of accuracy, the convolutional layer and a traditional one-class SVM algorithm.

In contrast to the above research, Lu *et al.* built an architecture explicitly modeling the temporal dependencies in bearing vibration data [70]. They used AE to extract features from vibration spectra; the learned features were organized sequentially to form a transition feature pool, which, in turn, was sent to an LSTM network. They compared their method to several alternatives and noted its superior effectiveness. The same pipeline was adopted by Li*et al.* in the same context [71].

Time-series data are relatively more complex than other data types, because they comprise temporal dependencies which need to be modeled either implicitly or explicitly. Jiang *et al.* proposed using the sliding window strategy to model temporal information in SCADA data to detect faults in wind turbines [72]. More specifically, they divided multivariate time-series data into fixed-length chunks along the time axis, allowing overlap between different chunks. They trained a DAE using chunks obtained under normal operating conditions. At the online stage, they applied the same sliding window strategy to test samples before sending them to the trained DAE; the residual error was the evidence of an anomaly. A nearly identical idea was adopted by Fan *et al.* to detect faults from building energy usage data [73]. Instead of slicing data along the time axis, Ellefsen *et al.* VAE model takes in one vector (multivariate measurements) at a time, and the chronological ordered residual errors form a new time series [74]. A high rate of increase in the new time series indicates a potential fault in maritime components.

Kim *et al.* took a different approach and proposed a model where temporal information was explicitly modeled in the architecture [75]. Their model, named DeepNAP, comprises two modules: prediction and detection. The former is essentially an AE with LSTM as its building blocks; it tries to predict a sequence of output which has a minimum reconstruction error with the given sequential input. The latter is a fully-connected MLP taking in only part of the output sequence from the previous step and projecting it to the remaining part of the sequence. With a newly suggested loss function, i.e., partial reconstruction loss, the two modules can be trained jointly; superior accuracy was observed when this was done. Similar work by Zheng *et al.* adopted AE with LSTM units, much like the above prediction module, to detect anomalous power demand [76]. To improve accuracy, robustness, and resistance to the spillover effect, Baraldi *et al.* compared the ability of RNN, auto-associative kernel regression and fuzzy similarity to detect faults from time-series temperature measurements. They further proposed an ensemble of these models and reported a satisfactory result [77].

Although imagery data have been extensively studied in fault detection applications, as shown in Fig. 7, unsupervised applications are rarely reported. The only example we found in the literature was an attempt by Kang *et al.* to detect defective catenary insulators [78]. They applied the Faster R-CNN algorithm to localize the insulator in an image first, and then built a denoising AE in an unsupervised manner. Unsurprisingly, at testing stage, reconstruction error was used as the evidence signifying a potential fault. To increase the representativeness of the feature embedded in the denoising AE, the authors enforced the encoder network to be shared with a different but similar task, classifying an image patch as an insulator or others, under supervision. In this way, the knowledge learned in one task could be transferred to another, i.e., multi-task learning, as done in [56].

The scarcity of unsupervised fault detection from imagery data may be attributed to the complexity of the data. Although, at times, the complexity may not be overly intimidating for human brains, image annotation is very labor-intensive and might hinder the replication of a model from one domain to another. We expect more studies to fill this research gap.

Structured data have been investigated in the regime of unsupervised fault detection as well. For example, Zhao *et al.* used SCADA data to build an AE to detect faults in a wind turbine [79]. Using only samples of normal condition, they conducted layer-wise pretraining and finetuning to train the network. They used the reconstruction error of a testing sample as an indicator of potential faults. To accommodate the non-stationary operating condition, they designed adaptive thresholds for triggering alarms following extreme value theory. Similar work used denoising AE to monitor sensors in a nuclear power plant [80].

Another common strategy when dealing with structured data is to select one target variable from the multivariate measurements and build a prediction model that maps all other measurements to this specified target. Notably, the target may not necessarily directly reflect the health status of the equipment of interest, but it should be dependent on other variables. Likewise, the prediction model should be trained with samples of normal condition. At the testing stage, given a set of incoming measurements, the residual error is the difference between the target prediction and the actual target measurement. This converts an unsupervised problem into a supervised one. Using this strategy, Wang *et al.* built a DBN for detecting faults in wind turbine using SCADA data, in which the main bearing temperature was selected as the target variable [81]. Other work by Wang *et al.* selected lubricant pressure as the target variable; in this case, the researchers built a feedforward neural network to detect faults in a wind turbine gearbox [82].

Fig. 7 provides a crosstab-like summary of the above fault detection applications, with a focus on the selection of deep learning models for different types of data input. The figure validates the universal applicability of deep learning for various data types. In general, all data require their intrinsic features to be extracted to better represent the patterns in negative samples, thus detecting positive samples which do not follow the patterns. Specifically, vibration data include time domain features, frequency domain features, time-frequency domain features and a combination of these; imagery data encompass spatial structural features; time-series data comprise temporal dependencies; structured data contain cross correlations. As shown in Fig. 7, the vast majority of deep learning models in the unsupervised regime are AE-based because their objective functions for training fit well with the learning mechanism, i.e., the use of reconstruction errors. While deep learning models are popular in supervised applications, a limited number of studies have considered unsupervised settings. This may be attributed to the attainability of data from laboratory tests and many easy-to-use supervised algorithms. Because of the high costs of obtaining labels in real-world applications, we expect more research on unsupervised fault detection using deep learning.

B. FAULT DIAGNOSIS

To return to our previous analogy, being aware of our own illness is not enough; we need to consult professionals to identify the type, localize the body part and identify the severity. By the same token, once an equipment fault is detected, steps need to be taken for fault recognition, fault localization and identification of severity, a process called fault diagnosis. The diagnosis procedure should be able to identify ''what went wrong'' (kind, situation and extent of the fault) as an extension of the knowledge that ''something went wrong'' derived at the previous step (i.e., detection). Fault diagnosis must be much more rigorous than fault detection in its prediction accuracy and results, since it may directly suggest the ensuing operation adjustments or maintenance tasks.

From a machine learning point of view, diagnosis is a multi-class classification problem, classifying a detected fault to a certain combination of fault type, location and severity. A typical design in the deep learning architecture is the addition of a soft-max layer to the final layer. Correspondingly, cross-entropy loss is often chosen as the loss function, based on which the network can be trained. After training a deep learning model, nonlinear dimensionality reduction methods, such as the t-SNE method, can be adopted to visually inspect whether the learned high-level features are discriminative; see [83] for an example. Typically used evaluation metrics include accuracy, precision, recall, ROC curve, AUC and F-score. A confusion matrix is often employed to visually investigate the classification results, especially to locate misclassified classes. Finding misclassifications may give a hint on the direction to take to improve the accuracy.

Another common property of fault diagnosis using deep learning is its use of supervised learning. Although feature representations can be pretrained in an unsupervised manner, their classifiers are mostly finetuned with labels. In these cases, we also consider them supervised. Though our review of the literature may not be exhaustive, all papers mentioned in the next subsections are under the supervised regime, like the supervised fault detection applications we have discussed. To make a difference from supervised fault detection (i.e., binary classification), we assume fault diagnosis have at least three different classes of fault types and the classes are more balanced. Similarly, we structure our analysis according to the type of input data.

1) VIBRATION DATA

Vibration or acoustic data are a significant, if not the most significant, source for diagnosing rotating and reciprocating machines whose health condition is critical to system safety and reliability. Given this, a large part of fault diagnosis research is about learning from vibration or acoustic signals. Depending on the integration level of the learning pipeline, we divide related work into two paradigms: separate learning and end-to-end learning, as shown in Fig. 8.

Like many conventional machine learning tasks, separate learning consists of several independent steps, including feature extraction, feature selection and pattern recognition. End-to-end learning builds an integrated network, taking in the raw signal, extracting discriminative feature representations and outputting the desired targets. The distinctions between these categories will be explained in the next section.

Data and features often determine the upper limit of learning performance, and models and algorithms only approximate this upper limit. In the following discussion, we intentionally neglect the finer details of how to construct a deep neural network and focus on data preparation and feature learning.

It is well-known that mechanical equipment faults can be easily concealed in time-domain waveforms and manifest better in the frequency domain. Numerous studies have adopted signal processing techniques. The most popular one, Fast Fourier Transform (FFT), is used to extract the frequency spectrum from original time-domain signals and forward them to a deep neural network [84]–[91]. However, the process might not work well for transient or stationary signals whose frequency components vary in time, usually the case in the real world. For non-stationary signals, it is common to transform the raw signals into the time-frequency domain using Short Time Fourier Transform (STFT) [92]–[94], Wavelet Transform (WT) [95]–[98] or Empirical Mode Decomposition (EMD) [99], [100]. STFT adopts a window function of fixed length, thus suffering from the time-frequency resolution trade-off problem. To improve

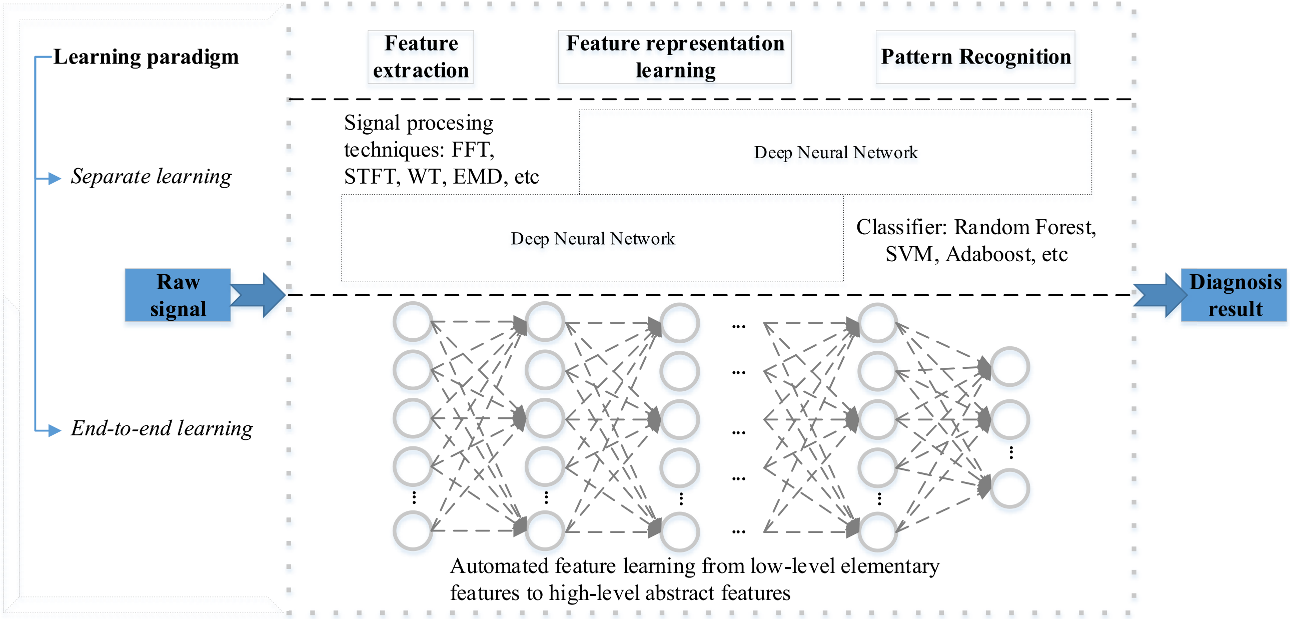

Learning paradigm
Feature extraction
Feature representation learning
Pattern Recognition
Separate learning
Signal procesing techniques: FFT, STFT, WT, EMD, etc
Deep Neural Network
Deep Neural Network
Classifier: Random Forest, SVM, Adaboost, etc
Raw signal
Diagnosis result
End-to-end learning
Automated feature learning from low-level elementary features to high-level abstract features

FIGURE 8. The two learning paradigms: separate learning vs end-to-end learning.

STFT, WT conducts multi-resolution analysis using varying window size for every single spectral component; however, it is very dependent on the basis function chosen and has shift-variant and poor directionality problems. EMD does not rely on any basis function, but it may suffer from the mode mixing problem in cases of intermittence and noise. Verstraete *et al.* compared the above three signal processing techniques and observed that WT yielded consistently high accuracy in their study [101]. To improve prediction accuracy, several researchers tried to fuse all the statistical features, also called tri-domain features, derived from the signal processing techniques mentioned above [102]–[105]. Note that signals in the time-frequency domain naturally have a two-dimensional form, making them suitable for the input of a traditional CNN. After a proper normalization step, those time-frequency domain signals were treated as images and various CNN-based variants were built by [92], [94]–[98], [101], [106], [107].

The raw vibration signal has also been transformed to an imagery form for diagnosis purposes using techniques like Continuous Interleaved Sampling (CIS) [108]–[112], Omnidirectional Regeneration Technique (ORT) [113] and Symmetrized Dot Pattern (SDP) [114], [115]. Armed with these transformation techniques and the transfer learning strategy, several pretrained CNNs, originally trained on natural images, were transferred to fault diagnosis applications using vibration data; examples include LeNet-5 [107], [109], [110], VGG-16 [106], AlexNet [95] and ResNet-50 [108]. We also found some studies using auto-encoder [105], [116], [117] and random projection [118] as a pre-posed layer before a deep neural network for the purpose of denoising and compressing.

Vibration data are essentially time series, but with strong periodicity and high sampling frequency. This sequential type of input can be fed into the input layer of an RNN or a 1D CNN, giving the opportunity to conduct end-to-end learning. Taking raw vibration signals as input, researchers built a standard RNN [119], [120] and 1D CNN [121], [122] to automatically learn representative features and output the desired targets. Some merits of the end-to-end learning paradigm are the following: it lets the data speak; feature engineering is automated, without the need for hand-crafted features; parameters of the whole network can be jointly optimized, leading to better accuracy; the network is generic and can be easily transferred or adapted to a different but similar scenario.

When adopting 1D CNN to directly process vibration data, the kernel width parameter should be designed with caution. With a narrow kernel width, the time resolution is better but the frequency resolution is poorer and vice versa. This is consistent with the support of the window function used in STFT. Peng *et al.* [123] and Zhang *et al.* [124] proposed using wide kernels to enhance the learning of low frequency fault-related features and suppress noise interference. In other work, a multi-scale 1D CNN [83], a dilated convolutional layer [125] and a combination of the LSTM layer with 1D CNN [126] were proposed for the same purpose. The stride parameter should also be selected carefully, because an overly large stride parameter will inevitably produce undesired shift-variant features. To improve accuracy, several studies attempted to fuse features produced by multiple 1D CNN [127] and GRU [128]. In the end-to-end learning regime, although rarely reported, AE-based models for vibration data have been investigated; see [129]–[131].

TABLE 1. Diagnostic research referring to the CWRU dataset.

Learning Paradigm	Reference	Model	Accuracy	Reference	Model	Accuracy
End-to-end learning	Zhang et al. [124]	1D CNN	100.00%	Wang et al. [131]	AE	99.63%
	Pan et al. [126]	1D CNN+LSTM	99.40%	Xia et al. [133]	CNN	99.41%
	Li et al. [127]	1D CNN	95.76%	Yu et al. [119]	LSTM	94.00%
	Jia et al. [130]	AE	99.92%			
Separate learning	Hoang et al. [111]	CIS+CNN	100.00%	Zhou et al. [84]	FFT+AE	99.79%
	Wen et al. [108]	CIS+CNN	99.99%	Jia et al. [88]	FFT+AE	99.61%
	Xu et al. [110]	CIS+CNN	99.88%	Shao et al. [106]	CWT+CNN	98.80%
	Wen et al. [109]	CIS+CNN	99.79%	Guo et al. [96]	CWT+CNN	97.79%
	Lu et al. [112]	CIS+CNN	92.60%	Ma et al. [95]	Frequency slice wavelet Transform + CNN	99.89%
	Nguyen et al. [100]	EMD+AE	100.00%	Wen et al. [134]	Auto-correlation power spectrum + AE	99.82%
	Jiang et al. [99]	EMD+AE	96.36%	Sohaib et al. [102]	Tri-domain features + AE	99.75%
	Verstraete et al. [101]	WPT+CNN	99.50%	Wang et al. [135]	CNN+HMM	98.13%
	Xu et al. [107]	WPT+CNN	99.08%	Sun et al. [118]	Random projection + AE	97.47%
	Ma et al. [98]	WPT+CNN	97.43%	Jiang et al. [136]	FFT+LSTM	94.75%

Interestingly, in the reviewing process, we discovered that most of the cited papers were referring to the Case Western Reserve University (CWRU) dataset [132], the de facto benchmarking dataset for rolling bearing fault diagnosis. The vibration data were collected under different faulty conditions, each of which was induced with a specific type of defect (inner race, roller, outer race) and one severity level (0.007, 0.014 and 0.021 inches). A typical problem formulation using the CWRU dataset is a ten-class classification considering the combination of various faulty types and severities together with the normal condition. As shown in Table 1, the testing accuracy of all the referenced papers is above 92%, with several surpassing 99%. Admittedly, this is astonishing given the large number of faulty conditions to classify.

Note that in Table 1, the most commonly used architectures for this dataset are CNN and AE. Also note that all the figures in Table 1 were excerpted from the original papers without verification. Though some researchers adopted the same model, they reported different testing accuracies. This may be attributed to such factors as the partition of the training and testing sets, hyperparameter tuning and random variation.

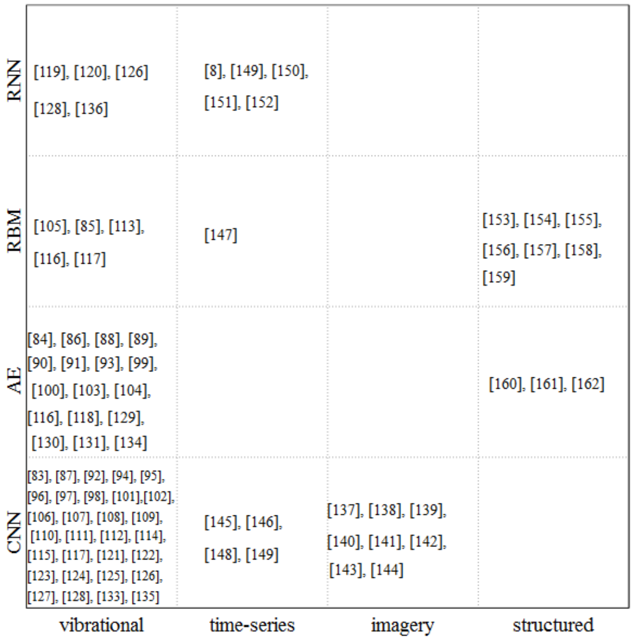
Quantitatively, the number of papers employing the end-to-end learning paradigm is still much less than the separate learning paradigm. With a significance level of 5%, we conducted a two-sided Welch's t-test on the difference of mean testing accuracy between the two learning paradigms and obtained a p value of 0.64. This implies there is no significant difference between the testing accuracy of the two learning paradigms. While this proves the traditional signal processing techniques can extract useful features from vibration data, the end-to-end learning paradigm should be favored as it does not require much human intervention in feature engineering.

2) IMAGERY DATA

Image classification research has progressed tremendously with the recent advancements in deep learning theory, especially the development of CNN. One seemingly intimidating obstacle to the application of fault diagnosis from imagery data is the availability of sufficiently labeled samples. However, this is generally not a prerequisite. With only 40 infrared thermal videos, each 10 minutes long, Janssens *et al.* successfully conducted rolling-element bearing fault diagnosis by reusing the well-known pretrained VGG-16 model [137]. They simply replaced the last layer of the architecture with a soft-max layer, restricting the number of nodes to the desired output classes, and finetuned the whole network with a limited number of samples. Although their application context differed significantly from the scenario on which the VGG-16 was originally trained, the knowledge learned was transferrable, leading to an accuracy of more than 95%. This type of transfer learning can markedly reduce the required number of labeled samples. Using the same strategy, Xu *et al.* adopted and compared four pretrained networks (SqueezeNet v1.1, Inception v3, VGG-16 and ResNet-18) in the context of roller bearing surface defect diagnosis [138]. They validated a gain in both convergence speed and accuracy using a pretrained network.

Transfer learning is not a panacea for all domains, however, and training from scratch can also yield reasonably good results. Tao *et al.* trained a compact CNN from scratch with only 50 raw images to diagnose metallic surface defects [139]. With the aid of data augmentation (including random rotation, translation, zoom, shear and elastic transformation) and a segmentation step prior to classification, they successfully augmented the number of labeled training samples. As a result, their model achieved an accuracy of 86.82%, much higher than classical models based on hand-crafted features. Similar work used data augmentation to diagnose furnace combustion states [140], weld flaw types [141] and balancing tail ropes' faults [142].

Depending on the complexity of a concrete problem, using a pretrained network or data augmentation techniques may not be necessary. Jia *et al.* trained a CNN with only



	vibrational	time-series	imagery	structured
RNN	[119], [120], [126] [128], [136]	[8], [149], [150], [151], [152]		
RBM	[105], [85], [113], [116], [117]	[147]		[153], [154], [155], [156], [157], [158], [159]
AE	[84], [86], [88], [89], [90], [91], [93], [99], [100], [103], [104], [116], [118], [129], [130], [131], [134]			[160], [161], [162]
CNN	[83], [87], [92], [94], [95], [96], [97], [98], [101],[102], [106], [107], [108], [109], [110], [111], [112], [114], [115], [117], [121], [122], [123], [124], [125], [126], [127], [128], [133], [135]	[145], [146], [148], [149]	[137], [138], [139], [140], [141], [142], [143], [144]	

FIGURE 9. Fault diagnosis applications: selection of deep learning models given different types of data input.

450 images per class to diagnose nine faulty states of rolling bearings, attaining nearly 100% accuracy [143]. Likewise, Li *et al.* trained a CNN with only 1400 samples per class to diagnose five module defects in photovoltaic farms, again with highly accurate results [144]. As shown in Fig. 9, all the above studies used CNN, with some differences in their network depth, the choice of regularization methods or training details.

3) TIME-SERIES DATA

Time-series data encapsulate temporal dependencies that are typically crucial for fault diagnosis. In a multivariate case, they may also contain cross correlations amongst multiple measurements, as in structured data. A naive assumption that samples at different timestamps are independent would discard the useful temporal information, thus inevitably leading to poor performance. Researchers have attempted to tackle this problem at two different levels: the data level and the algorithm level.

Data-level methods use phase space embedding representation in which a sequence of data instances is generated via a fixed-length sliding window; see [145]–[148]. Through data-level transformation, temporal dependencies may be translated into cross correlations. However, the determination of the window size and the sliding stride size becomes a problem which may require prior knowledge or extra efforts in hyperparameter tuning. Note that the stride size parameter may affect the shift-invariant property of the method, and this is sometimes desired for time-series data.

Algorithm-level methods explicitly model the temporal dependencies in their architectural design, mainly RNN. Examples of this type can be found in [8], [149]–[152]. Although the length of data input in an RNN needs to be determined beforehand, it is significantly different from the window size in data-level methods. RNN can learn the length of temporal dependencies via its memory retention mechanism, and the length is changeable, providing more flexibility in fault diagnosis using time-series data.

4) STRUCTURED DATA

Structured data have always been an important part of conventional fault diagnosis applications using machine learning. In the literature, the most commonly adopted deep learning architectures for structured data are RBM-based and AE-based, as shown in Fig. 9, possibly because these two types of model do not impose topological or sequential relationships when learning from input, unlike CNN and RNN. Instead, their architecture resembles a feedforward neural network, allowing cross correlation in the input to be learned. As explained in Section II, RBM-based and AE-based models can be trained in two steps: layer-wise pretraining, and fine-tuning on the network by stacking previously learned layers. Using this strategy, several researchers built regular DBNs [153]–[159] and AEs [160]–[163] for fault diagnosis; [153]–[155], [160] emphasized network hyperparameter tuning. Note that the layer-wise pretraining step is typically unsupervised, and the pretrained network serves as an initialization to the whole model. This can greatly reduce the number of labeled data required and boost the convergence speed [162].

Other researchers have endeavored to improve the classification accuracy by combining deep learning models with other models, such as the multi-grained cascade forest [164], fisher discriminative dictionary learning [165] and deep quantum neural network [166]. One commonality is they all used the deep learning model to learn feature representations and the other model to increase discriminative power.

Real-world structured data may originate from all sorts of sources, including current, voltage, speed, displacement, pressure, temperature and many others, and data fusion may be necessary [167]. Data may also be subject to problems like incompleteness, heterogeneity, low signal to noise ratio, exhibition of certain topology, etc. Chen *et al.* attempted to conquer the incomplete data problem caused by multi-rate sampling by using transfer learning [168]. They proposed a framework enabling a portion of the structure and parameters to be transferred between the model of structurally complete data and the model of incomplete data. An interesting study by Wang *et al.* proposed using CNN to tackle structured data with spatial topology embedded in them [169]. Their data contained power flow measurements in a power system, and the purpose was to diagnosis the system's faults. They designed rules transforming the original power flow into images, in which geometry, digits and other characteristics could be preserved. Experiments proved the efficacy of their model.

Fig. 9 presents a crosstab-like summary of the diagnosis studies reviewed above, with a focus on the selection of deep learning models for different types of data input. The figure reconfirms the universal applicability of deep learning to various data types. The figure may also serve as a dictionary for researchers to link future studies to existing ones.

Although numerous studies have validated the superiority of adopting deep learning in fault diagnosis, they are generally restricted to laboratory data, largely because of the insufficiency of labeled data in real-world applications where destructive experiments are costly or not allowed. Furthermore, the learned classifier may only be sensitive to those faults that are included in the training set. In other words, its generalization capability to unhappened faults may be poor, leading to low testing accuracy in the real world. For instance, a soft-max layer in a deep learning model outputs a fault type associated with a neuron with the highest activation, regardless of whether the fault pattern has been observed or not [93]. Strictly speaking, compound faults (several faults occur simultaneously) that are not included in the training set should also be considered unhappened. For this reason, several studies meticulously collected compound faults related data and incorporated them into their training set [104], [129], [143], [162]. However, this type of study is restricted in the sense that the combinatorial explosion of many faulty types prevents us from collecting sufficiently labelled data to train an all-in-one diagnostic model. A potential solution is to exploit an unsupervised fault detection model and periodically update the diagnostic model based on newly observed data samples.

C. PROGNOSIS

After diagnosing a disease, health professionals infer the patient's recovery or survival rate based on empirical data and their experience. In PHM, this is generally known as prognosis. In prognosis, we estimate the Remaining Useful Life (RUL) of the item of interest. This step projects the states of the monitored item into the future using a combination of prognostic models and future operational usage models. In other words, it estimates the RUL of the item taking into account its degradation trajectory and the future operational use plan. From a practical perspective, it is important to have an accurate RUL estimation, because an early prediction may result in over-maintenance and a late prediction could lead to catastrophic failures. With an appropriate RUL estimation, maintenance work can be adequately scheduled considering the required maintenance personnel, spare parts, tools and other logistics. In light of the uncertainties in the real world, a confidence level of the assessment is required to quantify the fluctuation in the RUL estimates.

From a machine learning viewpoint, prognosis is a regression problem, as the target value (RUL) is in the real domain. Prognosis aims to learn a function that maps the condition of an item to its RUL estimates. As in many regression tasks, it is challenging to provide labels for training. Specifically, in prognostic applications, it is hard, sometimes impossible, to accurately determine the RUL of an object at any given time. Most research uses data from run-to-failure tests, from which the RUL labels can be derived. The criteria defining a failure occurrence are application-dependent; for example, a machining tool is defined failed when its wear size achieves a threshold of 0.6mm [170], a lithium-ion battery fails when it has 30% capacity fade from the rated capacity [171] and a rotating bearing fails when its maximum vibration amplitude exceeds 20g [172].

The simplest way to define RUL is by calculating time to failure, i.e., subtracting the timestamp of the failure occurrence from each time step; see [173], [174] for examples. However, this inadvertently implies that the health state of an item degrades linearly with its usage and may result in over-estimation of the RUL. In some cases, a reasonable assumption is that the degradation of the monitored item is much less significant at the early stage of its lifecycle, and it starts to degrade only after a certain amount of usage. This yields a piece-wise linear setting of RUL, namely a constant RUL followed by a linear degradation function. The time point segmenting the piece-wise function can be set according to prior knowledge, as in [175]–[179]. It can also be determined via a fault detection procedure, using, for example, statistical process control [180], SVM [181], variational AE [182] or a singular value decomposition (SVD) normalized correlation coefficient [183]. As an alternative to the linearly decreasing function, researchers investigated power functions [181] and low-order polynomials [182] with the hope of better capturing the degradation pattern. Their findings verified the necessity of conducting fault detection before prognostic tasks. In general, fault detection techniques can facilitate the labeling of RUL in prognostic tasks.

In prognostic tasks, the final layer of a deep learning architecture can be a single neuron with a linear activation function [170], [184]–[186] or a sigmoid function squashing the RUL prediction to a normalized range [187], [188]. Accordingly, many loss functions can be selected for training; typical ones are Mean Absolute Percentage Error (MAPE) [174], Mean Absolute Error (MAE) [180] and Mean Squared Error (MSE) [181]. These loss functions can also be applied to evaluate model performance in a testing set.

It has to be noted that one unique characteristic of prognostic tasks is the penalization of late RUL predictions (i.e., the estimated RUL is larger than the actual RUL). Late prediction may lead to unplanned breakdown, or even catastrophic damage, whereas early prediction only causes extra maintenance cost. To cope with this problem, the following asymmetric scoring function for evaluating model performance was proposed by [189], adopted by [182], [190], [191] and modified by [192]–[194]:

$$s = \sum_{i=1}^{m} s_i,$$
$$s_i = \begin{cases} e^{-d_i/a_1} - 1, \text{if } d_i < 0 \\ e^{-d_i/a_2} - 1, \text{if } d_i \geq 0, \quad \text{where } a_1 > a_2 > 0 \end{cases} \tag{1}$$

where m is the number of testing samples, d_i equals to $\text{RUL}_{\text{estimated}} - \text{RUL}_{\text{actual}}$, denoting the difference between the estimated RUL and the actual RUL of the i-th sample, and the magnitude of a_1 and a_2 controls the degree of penalty for late predictions. Though reasonable, the use of a scoring function tends to underestimate RUL values, which may or may not coincide with the user's intention. The exponential form of the scoring function also makes it extremely sensitive to outliers. In other words, a very bad prediction can dominate the overall score, masking the accuracy of other predictions. This makes the selection of evaluation metric very application-dependent.

Having introduced the fundamentals of prognostic tasks, we now provide a detailed analysis of existing work. In the literature, related papers we found are extremely imbalanced in their types of input data. Therefore, the following analysis is structured according to their application scenarios instead of data types. The most researched items for RUL prediction are the machining tool, battery, turbofan engine and rotating bearing, with the last two mainly referring to the benchmarking datasets from the PHM 2008 data challenge [195] and IEEE PHM 2012 data challenge [196].

Although the wear of a machining tool can be measured offline, it is desirable to monitor and predict tool wear in real time using online measurements, typically force and vibration signals. zadeh *et al.* proposed a spectral subtraction method to intensify fault signatures by subtracting the WPT spectrum of a signal by its steady-state part; the obtained residuals were fed to a standard CNN for tool wear estimation [186]. To better model the degradation trend, Wang *et al.* proposed using bidirectional GRU to capture the temporal-dependencies among the tri-domain features of the original signal [185], while Rui *et al.* used CNN for feature extraction and bidirectional LSTM for sequential modeling [184]. Comparisons of traditional machine learning and deep learning models, including CNN, LSTM, AE and DBN, in tool wear prediction can be found in [197].

Battery RUL prediction is of great practical significance in modern life given the ubiquity of portable equipment, but the complex electrical-chemical nature of the battery makes it difficult to use first principles to model its degradation mechanism. Data-driven methods attempt to learn a function by mapping multivariate time-series measurements of a battery (current, voltage, temperature, etc.) to its capacity retention, a common indicator signifying the life of a battery. The use of deep learning in battery prognosis is still in its early stages. There is a limited amount of work in the literature; for example, feedforward DNN [171], [198] and regular LSTM [199], [200] have been used in studies as the function approximator.

The PHM 2008 data challenge asked researchers to predict the RUL of NASA's turbofan engines based on multivariate time-series measurements, also known as the C-MAPSS dataset [195]. The dataset is comprised of four sub-datasets subjecting to different operating and fault conditions. Several regular deep learning models, such as sparse AE [201], CNN [202], LSTM [177], [181], were used to tackle the problem. Using bidirectional LSTM, Zhang et al. studied the transferability of the problem among different operating conditions [191]. An interesting observation was that negative transfer occurred when transferring from a dataset of multiple operating conditions to a dataset of single operating condition. To improve accuracy, Long et al. built a k-fold ensemble model using residual CNN; this method was similar to the bagging technique in machine learning. In a method resembling the principle of random forest, Zhang et al. constructed a multiple DBN ensemble to maximize two conflicting objectives: accuracy and diversity. Composite models using LSTM with RBM [190] and 1D CNN [176], [182] were also investigated recently, and quite competitive performance was reported.

In one of the most researched prognostic problems, the IEEE PHM 2012 data challenge works with bearing vibration data acquired from an accelerated aging platform PROGNOSTIA. It expects participants to predict the RUL of bearings whose monitoring data are truncated [196]. Although the target is different, the data type is in line with the aforementioned fault diagnosis problem using the CWRU dataset. Therefore, we provide some details of related studies in Table 2 with the aim of comparing them to those in Table 1. Although we tried to provide a unified metric for better comparison, different people use different metrics when evaluating model performance. Note that the ''Evaluation'' column of Table 2 has three different metrics, i.e., Score, RMSE and MAPE, and their magnitudes are incomparable. In addition, the score is different from that defined in equation (1). The higher the score the better the accuracy; see [196] for a definition. As in Table 1, the figures in Table 2 are directly from the original paper or indirectly calculated and have not been verified.

After carefully scrutinizing these studies, we found the proposed methods share many common properties with those in Table 1. A simple modification by replacing the final classification layer with a regression layer can turn a diagnostic method into a prognostic one. This observation was also made by [183] and [185]. This gives an opportunity to conduct transfer learning in these two closely related but different tasks. Another interesting observation was that all the proposed methods in Table 2 fall into the category of separate learning, not end-to-end learning. In light of all the merits of end-to-end learning, we expect more studies of this type to work on prognostic problems in the future.

This subsection surveys prognostic research that aims to predict the RUL of machining tools, batteries, turbofan engines and rotating bearings. After reviewing the referenced papers, we made some interesting observations:

1) Compared with fault detection and diagnosis, no imagery data were used as input in prognosis tasks. This may imply that degradation process is not evident in images in some domains. It may also indicate the potential to develop imagery data-driven prognostic applications.
2) While a confidence bound associated with the target RUL prediction is a desirable output, very few

TABLE 2. Comparison of prognostic research referring to the IEEE PHM 2012 data challenge.

Ref.	Evaluation	Model	Main steps of the proposed method
Liao et al. [193]	Score: 0.588	RBM	1. RBM was used for feature extraction. A slope term was added to the loss function encouraging the features to learn trendability embodied in the degradation pattern; 2. The learned features were then fed to a self-organizing map for RUL prediction.
Yoo et al. [192]	Score: 0.5689	CNN	1. CWT was used to transform raw vibration signal to time-frequency image features; 2. CNN (LeNet-5) was constructed to extract spatial features from the images, and to map to the bearing's health index.
Guo et al. [187]	Score: 0.5655	LSTM	1. Tri-domain features were firstly extracted from raw vibration signal; 2. A feature selection method based on correlation and monotonicity was used to further select degradation-sensitive features; 3. A LSTM was constructed to map from the selected features to the bearing's health index.
Wang et al. [188]	Score: 0.5098	CNN	1. Fourier transform was used to extract frequency domain features from vibration signal; 2. Continuous interleaved sampling was used to convert the features to images; 3. CNN(LeNet-5) was constructed to extract spatial features from the images, and to map to the bearing's degradation percentage.
Zhu et al. [194]	Score: 0.3624	CNN	1. Wavelet transform was used to extract time-frequency domain spectrum; 2. The bilinear interpolation method was used for dimensionality reduction; 3. The derived low-dimensional features were fed to a CNN, and features from different layers were concatenated prior to RUL prediction.
Ren et al. [172]	RMSE: 0.0414	DNN	1. Fourier transform was used to extract frequency domain features of vibration signal, then a method called Frequency Spectrum Partition Summation was used for dimension reduction; 2. Together with time domain features, the above features were fed to a DNN for RUL prediction.
Ren et al. [203]	RMSE: 0.119	CNN	1. FFT was used to extract frequency spectrum from vibration signal, and a one-dimensional spectrum principal energy vector (SPEV) was obtained via subsampling; 2. SPEV of consecutive time steps were stacked to form a matrix, which was then fed to a CNN for RUL prediction.
Ren et al. [204]	RMSE: 0.152	GRU	1. Tri-domain features were firstly extracted, and they were compressed via an RBM; 2. With different time scales, the above features were fed to different GRUs, and the output of which were concatenated to densely connected layers for RUL prediction.
Li et al. [180]	RMSE: 0.236	CNN	1. Statistical process control based on kurtosis was used for incipient fault detection; 2. After detecting a fault, STFT was used to extract time-frequency domain spectrum from vibration signal; 3. The above spectrum was then fed to CNN, and all the learned features in different convolutional layers were concatenated for RUL prediction.
Ren et al. [205]	RMSE: 0.4472	AE+DNN	1. Tri-domain features were firstly extracted from raw vibration signal; 2. To reduce complexity, time-domain features were compressed via deep AE; 3. All the features were concatenated and fed to a feedforward DNN for RUL prediction.
Mao et al. [183]	MAPE: less than 0.5%	CNN+LSTM	1. Hilbert-Huang-Transform was used for feature extraction; 2. CNN was used to map the above time-frequency domain features to a binary healthy state: normal or faulty. The healthy state labels were obtained based on SVD correlation coefficient; 3. High-level feature representations of the CNN were further sent to LSTM for RUL prediction

researchers handle the requirement properly or report their efforts sufficiently. This should be addressed in future studies.

3) A few benchmarking datasets, such as the C-MAPSS and the PROGNOSTIA, are heavily used for the purpose of model validation. However, researchers tend to use different metrics to evaluate their models' accuracy, making comprehensive comparisons difficult. We call for a unified evaluation metric for model assessment in future studies.

Many studies explicitly encode temporal information of the sensor measurements using 1D CNN, RNN and their variants, while others implicitly encode them using the sliding window strategy. Researchers agree on the importance of capturing the temporal dependencies of data in prognostic tasks.

IV. CHALLENGES AND OPPORTUNITIES

A. CHALLENGES

In the fourth industrial revolution, or Industry 4.0, a key objective is to upgrade equipment's ability to perceive its own health state and predict future behavior. The development of PHM theory and practice aligns with this objective. As we have shown in the preceding sections, many pilot studies indicated deep learning is a promising tool in facilitating PHM applications. But this cross-disciplinary research is challenging, and the way ahead is long and arduous. Based on the work surveyed so far, in this section, we summarize some issues that have been overlooked or insufficiently dealt with to date. We also point out some challenges facing future applications of deep learning in PHM. Notably, however, the following challenges may not be unique to PHM. Some of the challenges share commonality with deep learning applications in other fields.

First, the use of deep learning is still an art. It requires experienced practitioners to select an appropriate deep learning model, regularize a too-complex model to prevent overfitting, pick a proper learning rate for faster convergence, tune the hyperparameters so the model has better generalization capability, consider scalability in a big data environment, and many others. All these requirements apply to the scenario of PHM applications; for example, as shown in Table 1, the use of the same model to solve the same problem yielded different testing accuracies. Although deep learning is known for its automated feature learning capability which alleviates the need for domain-specific knowledge in feature engineering, the above requirements pose yet another difficulty when adopting deep learning technologies. The solution relies on the further development of deep learning theory. It also demands a better documentation of best practices in PHM using deep learning.

Second, most research mentioned here conducted model validation on datasets gathered from bench-scale experiments, and this leads to poor generalization to real-world applications. Laboratory experiments attempt to simulate reality but often with simplifications or strong assumptions that may not hold in reality. Assessment of the health condition of an in-situ item is complicated, because it may be affected by too many factors, such as operating conditions, intercorrelations, multi-sourced and heterogeneous data, noises etc. Another concern is the scarcity of labeled samples in the real world. While fault injection and run-to-failure lifetime testing may be allowed in laboratory tests, destructive experiments are typically restricted in the real world for safety and economic reasons, resulting in insufficient labels. In spite of the challenges, we believe transfer learning is a bridge that can link laboratory tests and the real world, as will be detailed later in this section. In general, applying deep learning in real-world PHM applications is more complex than in labs, but we expect more studies to employ in-situ data in the future.

Third, an important but relatively understudied aspect is the ''concept drift'' problem, also known as covariate shift, in nonstationary data streams. In other words, online data may have a time-varying characteristic causing the model trained in an offline stage to become obsolete over time. This is generally true for any real-world PHM applications but the impact is less significant in rigorously controlled laboratory tests. Concept drift can sometimes be partly explained by contextual information, such as the load, rotating speed, ambient temperature etc. Therefore, it is necessary to incorporate contextual information as input to the deep learning model. Alternatively, concept drift may result from an item's intrinsic mechanism and contingent factors, making it unavoidable. To prevent deterioration in prediction accuracy because of concept drift, both active and passive solutions are appropriate. The former type relies on explicit change-detection and retrains the deep learning model after the detection of concept drifts. The latter type designs the model to be self-adaptive by adjusting the network parameters upon the arrival of new samples. Each solution has advantages and disadvantages, so they should be investigated individually for specific PHM scenarios.

Fourth, timeliness is a primary concern in PHM applications. An accurate but late prediction of a fault occurrence may not allow adequate time for remedies, causing damage or losses. This requires on-the-fly data processing and low-latency responses. However, current research makes little mention of time complexity analysis. Deep learning algorithms are typically computationally demanding, and some may rely on the computing power of Graphical Processing Units (GPUs). A conventional scheme of deep learning applications resorts to cloud computing, where massive amounts of data are transmitted to the cloud for computation, and the results are transmitted back to end users. This scheme may suffer from the problem of limited bandwidth and delayed response. To this end, edge computing has been proposed to bring computation and storage closer to the location where it is needed. The new scheme exploits field computing resources and can improve response times and save bandwidth. The deployment of edge computing for PHM is another related challenging task.

Last but not least, as shown in Fig. 1, an ideal PHM solution should output actionable tasks after identifying faults and estimating the RUL of the monitored system. The actionable tasks may be operational adjustments or prescriptive maintenance. Decision theory comes into play here, taking into account factors like risk, mission criticality, life-cycle costs, resource constraints and cost-benefit balance. Theoretically, deep reinforcement learning can play a role in systems consisting of states, actions, and rewards. The ultimate goal is to maximize long-term rewards by recommending feasible actions based on current states. However, the reality is complex, as all the above factors may vary along the time axis. In addition, it is hard to get enough data, a collection of triplets, i.e., ''state-action-reward'', to train the deep reinforcement learning model. The credit assignment problem, assigning long-term rewards or losses to each individual action, is a final challenging task. To the best of our knowledge, no such studies have been performed in the PHM area.

B. OPPORTUNITIES

Fortunately, opportunities always accompany challenges. Data are raw material that can be repeatedly exploited to extract information, knowledge and wisdom. Industrial digitalization is accelerating the speed of data collection, enhancing data richness and increasing computing power. On the demand side, industries long for high reliability and safety – and this was the original intention of PHM. In light of the powerful representing capability and the universal applicability to various types of data, deep learning can serve as a tool to mine data to achieve the goal of PHM. Based on our review, in the following, we point out three technologically related innovations in data science as the opportunities that could make possible for the further betterment in PHM.

1) TRANSFER LEARNING

Transfer learning aims to take advantage of experience learned in a source problem to improve the learning of a target problem [206]. The major merits are threefold: it alleviates the demand for a large number of labeled samples; it can accelerate the convergence speed in model training; it can boost the prediction accuracy. Many studies have already demonstrated the efficacy of transfer learning in PHM; therefore, we believe it is a very promising tool for future studies.

However, cautions should be taken to prevent negative transfer as we discussed earlier in Section III-C, and one example of this can be referred to [191]. Depending on the domain (feature space and its marginal distribution) and task (label space and the predictive function) differences of the source and target, transfer learning can be categorized into three types: inductive, transductive and unsupervised [206].

Unsupervised transfer learning focuses on solving unsupervised target problems, when no labels are available in the source and target. Although fault detection can be realized in an unsupervised manner, there is little research on this topic.

Inductive transfer learning has different source and target tasks, and some labeled data in the target problem are required to induce the predictive function. As introduced in Section III-A, Gibert et al. and Kang et al. both built a deep multi-task learning model for the purpose of fault detection from imagery data, namely the target problem [56], [78]. In the two studies, the source problems were material classifications, and the annotation of material classes was found to be easier than that of fault types. This is thought-provoking, as in some cases, it is hard to annotate the data as desired but easy to annotate them in a different way, giving the opportunity for multi-task learning. Another example of a widely studied type of inductive transfer learning in PHM is the reuse of a pretrained network, such as LeNet, VGG, AlexNet, and ResNet; see Section III for more examples. Note that fault detection, diagnosis and prognosis may have the same type of input but different targets, as illustrated in Section III. Deep learning provides a general framework for these PHM tasks. An architecture designed for one task is transferrable to another with effortless modification, opening the window to transfer learning, though rarely reported in the literature. For example, we can reuse a deep neural network originally trained for diagnosis in a new scenario for prognosis simply by replacing the final soft-max layer with a regression layer and finetuning the whole network.

Transductive transfer learning has the same source and target task, but the domains are different. In addition, labels in the source problem are abundant while labels in the target problem are sparse. The differences between the source and target domains may exist in the feature space or the marginal distribution. The former depicts a greater difference, also known as domain adaptation; hence, it can better resemble the cases transferred from laboratory tests to real-world applications. Existing work of this type is rare. We believe the challenge in large-scale expansion of PHM applications in the real world can be met via transfer learning. The latter (differences in marginal distribution) represents a smaller difference, which can be regarded as sample selection bias or covariate shift. Examples of this type appear in [134], [191], where a source network was trained with labels in one operating condition and transferred to another with no target labels. The difference in operating conditions explained the difference in marginal distributions of input.

2) DATA AUGMENTATION

In general, the number of samples for training has a direct impact on the upper bound of a deep neural network's accuracy. However, data labeling is often tedious, labor-intensive and costly. We can use data augmentation techniques to obtain more training samples to improve the performance of PHM applications. For example, when methods like random crop, rotation, translation, zoom, shear and elastic transformation were adopted on natural images to generate more training samples for fault diagnosis, improved accuracy was reported [62], [124], [138]–[140], [142]. The success of this type of data augmentation on natural images is explained by the human visual perception mechanism; for instance, a rotated cat can still be recognized as a cat by the human brain. The photographing of natural images is subject to ambient light, focal length, canted angle etc., and this explains the validity of the augmented data. Other data augmentation methods requiring knowledge of the first principles of the system, such as high-fidelity simulation, are not within the scope of this paper.

As for other types of data, there is no such intuition to guide us to generate data that are more real. Fortunately, the rise and recent development of Generative Adversarial Network (GAN) provides a partial solution. GAN is a new type of deep learning framework which consists of two models: a generator and a discriminator [207]. The generator aims to generate synthetic samples so that it can fool the discriminator, while the discriminator tries to distinguish the generated fake data from real data. The two models are pitted against each other until the fake samples are indistinguishable from genuine ones. Using this framework, Shao *et al.* built an auxiliary classifier GAN, named ACGAN, to generate artificial vibration data for fault diagnosis [208]. In cases of class imbalance, the use of GAN-based data augmentation supplemented the minor classes and improved the accuracy. More examples of this appear in [209]–[212].

The above-mentioned data augmentation methods for natural images rely on domain knowledge, whereas GAN-based data augmentation is automated without human intervention. This is analogous to hand-crafted features using conventional signal processing techniques versus automated feature learning using deep learning. More importantly, with a proper model design, GAN-based data augmentation can synthesize any type of data, attesting to its universal applicability. Overall, while the study of GAN-based data augmentation in PHM applications is in its infancy, we believe there are great opportunities to use it in future studies.

3) END-TO-END LEARNING

As shown in Fig. 8, end-to-end learning refers to building an integrated network as a whole, taking in raw signals and directly outputting the desired targets. This is in contrast to separate learning, in which feature learning and pattern recognition are independent. The primary advantages of end-to-end learning are fourfold. First, it lets the data speak; it can capture whatever statistics are in the data rather than being forced to reflect human preconceptions. Second, feature engineering is automated, without needing hand-crafted features. This vastly reduces the degree of dependency on domain knowledge and lowers the barrier to PHM applications. Third, parameters of the whole network can be jointly optimized, typically improving accuracy. This can be compared to separate learning, where a global optimum might not be achieved by optimizing each individual learning stage. Fourth, the network is generic, and from an architectural perspective, it can be easily transferred or adapted to a different but similar scenario. This is consistent with what we claimed above: deep learning provides a general framework which increases the transferability of deep learning models in PHM applications.

Although many studies have proven the effectiveness of end-to-end learning, increasing model complexity will create a need for more labeled training samples. Dividing the learning pipeline into several steps might help, especially when labeled data are sufficient in each individual step but insufficient from an end-to-end perspective. For example, fault detection using imagery data was accomplished through separate learning – object location followed by object classification [62], [63]. In cases of limited training samples, separate learning can be a good option, as human preconception provides an auxiliary approach to feature learning. But end-to-end learning should be favored when enough labeled data are available because of the automation in feature learning. We also think hybrid models incorporating both hand-crafted features and automated learned features can further boost the performance of PHM applications.

Deep learning is noted for its capability in automated feature learning. However, one important but neglected fact is that the learned features are part of a larger ''black box''. We believe a large body of physical meanings are embedded in these features in PHM applications. There is a fundamental opportunity for research to demystify the underlying mechanisms. As shown by [137], CNN can be made interpretable, and new insights can be gained into the underlying physics. More examples appear in [121], [124]. In these studies, the automatically learned 1D filters (convolutional kernels) were visualized and found similar to the basis function used in signal processing; each focused on extracting one or more specific frequency component in the vibration signal.

V. CONCLUSION

Many areas have been or are being transformed by deep learning technologies, including financial fraud detection, medical image diagnosis, machine translation and so on. Because it is a data-intensive field, PHM research is also reaping benefits from the advancement of deep learning theory. Traditional PHM applications have a fairly high technical barrier, as they require human expertise in statistics, signal processing, domain knowledge and many other skills. The most attractive specialty of deep learning is the automation of feature learning without the need for supervision. This greatly reduces the height of the technical barrier of PHM applications.

Deep learning provides a one-fits-all framework for PHM applications: fault detection uses reconstruction error or stacks a binary classifier on top of the learned network to detect anomalies; fault diagnosis typically adds a soft-max layer to perform multi-class classification; and prognosis adds a continuous regression layer to predict remaining useful life. The selection of a concrete deep learning architecture is application-dependent; it mainly depends on the type of data available. The analysis in this paper may suggest how to select an appropriate deep learning architecture for a specific application scenario.

Problem-solving in PHM and the theoretical development of deep learning can be seen as parasite and host, forming a mutually beneficial mechanism. In the literature, an increasing amount of research is focusing on fault detection, diagnosis and prognosis using deep learning. This paper surveys this work, reveals some of the common properties, pinpoints some important but overlooked issues and indicates challenges and potential opportunities for future studies. We can anticipate more research and industrial applications using deep learning in the PHM domain in the near future.

REFERENCES

[1] P. W. Kalgren, C. S. Byington, M. J. Roemer, and M. J. Watson, ''Defining PHM, a lexical evolution of maintenance and logistics,'' in *Proc. AUTOTESTCON*, 2007, pp. 353–358.

[2] J. W. Sheppard, M. A. Kaufman, and T. J. Wilmering, ''IEEE standards for prognostics and health management,'' *IEEE Aerosp. Electron. Syst. Mag.*, vol. 24, no. 9, pp. 34–41, Sep. 2009.

[3] J. Lee, M. Ghaffari, and S. Elmeligy, ''Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems,'' *Annu. Rev. Control*, vol. 35, no. 1, pp. 111–122, 2011.

[4] K. Holmberg, A. Adgar, A. Arnaiz, E. Jantunen, J. Mascolo, and S. Mekid, ''Information and communication technologies within E-maintenance,'' in *E-maintenance*. Cham, Switzerland: Springer, 2010, pp. 39–60.

[5] *Condition Monitoring and Diagnostics of Machines—Data Processing, Communication and Presentation, Part 2: Data Processing*, Standard ISO 13374-2:2007, 2007.

[6] M. Schwabacher, ''A survey of data-driven prognostics,'' in *Proc. AIAA Infotech Aerosp. Conf.*, 2005, pp. 1–5.

[7] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, ''Prognostics and health management: A review on data driven approaches,'' *Math. Probl. Eng.*, vol. 2015, no. 6, pp. 1–17, 2015.

[8] J. Yang and J. Kim, ''An accident diagnosis algorithm using long short-term memory,'' *Nucl. Eng. Technol.*, vol. 50, no. 4, pp. 582–588, 2018.

[9] L. Zhang, J. Lin, and R. Karim, ''Adaptive kernel density-based anomaly detection for nonlinear systems,'' *Knowl.-Based Syst.*, vol. 139, pp. 50–63, Jan. 2018.

[10] J. Fan, K. C. Yung, and M. Pecht, ''Physics-of-failure-based prognostics and health management for high-power white light-emitting diode lighting,'' *IEEE Trans. Device Mater. Rel.*, vol. 11, no. 3, pp. 407–416, Sep. 2011.

[11] M. Pecht and J. Gu, ''Physics-of-failure based prognostics for electronic products,'' *Trans. Inst. Meas. Control*, vol. 31, nos. 3–4, pp. 309–322, 2009.

[12] Z. Gao, C. Cecati, and S. X. Ding, ''A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches,'' *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[13] X. Dai and Z. Gao, ''From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis,'' *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2226–2238, Nov. 2013.

[14] L. Zhang, J. Lin, and R. Karim, ''An angle-based subspace anomaly detection approach to high-dimensional data: With an application to industrial fault detection,'' *Rel. Eng. Syst. Saf.*, vol. 142, pp. 482–497, Oct. 2015.

[15] L. Zhang, J. Lin, and R. Karim, "Sliding window-based fault detection from high-dimensional data streams," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 2, pp. 289–303, Feb. 2017.

[16] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[17] P. Elod, M. Koppany, T. Levente, and B. Lucian, "Railway track following with the AR. Drone using vanishing point detection," in *Proc. IEEE Conf. Automat., Qual. Test., Robot.*, May 2014, pp. 1–6.

[18] A. K. Singh, A. Swarup, A. Agarwal, and D. Singh, "Vision based rail track extraction and monitoring through drone imagery," *ICT Express*, to be published.

[19] R. K. Singleton, E. G. Strangas, and S. Aviyente, "Extended Kalman filtering for remaining-useful-life estimation of bearings," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1781–1790, Mar. 2015.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[21] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, Jun. 2011.

[22] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring: A survey," *arXiv Preprint arXiv:1612.07640*, vol. 14, no. 8, pp. 1–14, 2016.

[23] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, vol. 5, pp. 17368–17380, 2017.

[24] L. Liao and F. Köttig, "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction," *IEEE Trans. Rel.*, vol. 63, no. 1, pp. 191–207, Mar. 2014.

[25] M. Cerrada, R.-V. Sánchez, C. Li, F. Pacheco, D. Pacheco, D. Cabrera, J. V. de Oliveira, and R. E. Vásquez, "A review on data-driven fault severity assessment in rolling bearings," *Mech. Syst. Signal Process.*, vol. 99, pp. 169–196, Jan. 2018.

[26] D. Wang, K.-L. Tsui, and Q. Miao, "Prognostics and health management: A review of vibration based bearing and gear health indicators," *IEEE Access*, vol. 6, pp. 665–676, 2017.

[27] P. Lall, R. Lowe, and K. Goebel, "Prognostics and health monitoring of electronic systems," in *Proc. 12th Int. Conf. Thermal, Mech. Multi-Phys. Simulation Exp. Microelectron. Microsyst.*, 2011, pp. 1–17.

[28] G. Zhao, G. Zhang, Q. Ge, and X. Liu, "Research advances in fault diagnosis and prognostic based on deep learning," in *Proc. Prognostics Syst. Health Manage. Conf.*, 2016, pp. 1–6.

[29] J. Snoek, R. P. Adams, and H. Larochelle, "Nonparametric guidance of autoencoder representations using label information," *J. Mach. Learn. Res.*, vol. 13, pp. 2567–2588, Jun. 2012.

[30] M. Ranzato, C. Poultney, S. Chopra, and L. Yann, "Efficient learning of sparse representations with an energy-based model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1137–1144.

[31] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Jan. 2010.

[32] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[33] G. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 599–619.

[34] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.

[35] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[36] K. Cho, T. Raiko, and A. Ilin, "Gaussian-Bernoulli deep Boltzmann machine," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–7.

[37] R. Salakhutdinov and G. Hinton, "A better way to pretrain deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2447–2455.

[38] L. Yann, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[40] R. Jozefowicz, Z. Wojciech, and S. Ilya, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv Preprint arXiv:1412.3555*, pp. 1–9, Dec. 2014.

[43] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[44] D. Hawkins, *Identification of Outliers*. London, U.K.: Chapman & Hall, 1980.

[45] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[46] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle, S. Van Hoecke, "Convolutional neural network based fault detection for rotating machinery," *J. Sound Vib.*, vol. 377, pp. 331–345, Sep. 2016.

[47] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *J. Sound Vib.*, vol. 388, pp. 154–170, Feb. 2017.

[48] M. Bach-Andersen, B. Rømer-Odgaard, and O. Winther, "Deep learning for automated drivetrain fault detection," *Wind Energy*, vol. 21, no. 1, pp. 29–41, 2018.

[49] L. Guo, Y. Lei, N. Li, and S. Xing, "Deep convolution feature learning for health indicator construction of bearings," in *Proc. Prognostics Syst. Heal. Manag. Conf. (PHM)*, Harbin, China, 2017.

[50] B. Luo, H. Wang, H. Liu, B. Li, and F. Peng, "Early fault detection of machine tools based on deep learning and dynamic identification," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 509–518, Jan. 2018.

[51] Q. P. Hu, M. Xie, S. H. Ng, and G. Levitin, "Robust recurrent neural network modeling for software fault detection and correction prediction," *Rel. Eng. Syst. Saf.*, vol. 92, no. 3, pp. 332–340, 2007.

[52] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using LSTM networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2017.

[53] O. Obst, "Distributed fault detection in sensor networks using a recurrent neural network," *Neural Process. Lett.*, vol. 40, no. 3, pp. 261–273, 2014.

[54] M.-F. Guo, X.-D. Zeng, D.-Y. Chen, and N.-C. Yang, "Deep-learning-based earth fault detection using continuous wavelet transform and convolutional neural network in resonant grounding distribution systems," *IEEE Sens. J.*, vol. 18, no. 3, pp. 1291–1300, Feb. 2017.

[55] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, Nov. 2016.

[56] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 153–164, Jan. 2017.

[57] Y. Santur, M. Karaköse, and E. Akin, "A new rail inspection method based on deep learning using laser cameras," in *Proc. Int. Artif. Intell. Data Process. Symp.*, 2017, pp. 1–6.

[58] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Dec. 2016, pp. 3708–3712.

[59] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, May 2017.

[60] F.-C. Chen and R. M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2017.

[61] D. K. Jha, A. Srivastav, and A. Ray, "Temporal learning in video data using deep learning and Gaussian processes," *Int. J. Progn. Heal. Manag.*, vol. 7, no. 022, p. 11, 2016.

[62] J. Chen, Z. Liu, H. Wang, A. Nunez, and Z. Han, "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 2, pp. 257–269, Dec. 2017.

[63] X. Lei and Z. Sui, "Intelligent fault detection of high voltage line based on the Faster R-CNN," *Measurement*, vol. 138, pp. 379–385, May 2019.

[64] L. Chen, G. Xu, Q. Zhang, and X. Zhang, "Learning deep representation of imbalanced SCADA data for fault detection of wind turbines," *Measurement*, vol. 139, pp. 370–379, Jun. 2019.

[65] S. Mandal, B. Santhi, S. Sridhar, K. Vinolia, and P. Swaminathan, "Nuclear power plant thermocouple sensor-fault detection and classification using deep learning and generalized likelihood ratio test," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 6, pp. 1526–1534, Jun. 2017.

[66] J. Sun, R. Wyss, A. Steinecker, and P. Glocker, "Automated fault detection using deep belief networks for the quality inspection of electromotors," *Technisches Messen (TM-TECH MESS)*, vol. 81, no. 5, pp. 255–263, 2014.

[67] D. Y. Oh and I. D. Yun, "Residual error based anomaly detection using auto-encoder in SMD machine sound," *Sensors*, vol. 18, no. 5, pp. 1–14, 2018.

[68] Y. Park and I. D. Yun, "Fast adaptive RNN encoder–decoder for anomaly detection in SMD assembly machine," *Sensors*, vol. 18, no. 10, pp. 1–11, 2018.

[69] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA J. Autom. Sinca*, vol. 6, no. 2, pp. 441–451, Mar. 2019.

[70] W. Lu, Y. Li, Y. Cheng, D. Meng, B. Liang, and P. Zhou, "Early fault detection approach with deep architectures," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 7, pp. 1679–1689, Jul. 2018.

[71] Z. Li, J. Li, Y. Wang, and K. Wang, "A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment," *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 1–4, pp. 499–510, 2019.

[72] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using denoising autoencoder with temporal information," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 89–100, Feb. 2018.

[73] C. Fan, F. Xiao, Y. Zhao, and J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Appl. Energy*, vol. 211, pp. 1123–1135, Feb. 2018.

[74] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, and H. Zhang, "An unsupervised reconstruction-based fault detection algorithm for maritime components," *IEEE Access*, vol. 7, pp. 16101–16109, 2019.

[75] C. Kim, J. Lee, R. Kim, Y. Park, and J. Kang, "DeepNAP: Deep neural anomaly pre-detection in a semiconductor fab," *Inf. Sci.*, vols. 457–458, pp. 1–11, Aug. 2018.

[76] F. Zheng, S. Li, Z. Guo, B. Wu, S. Tian, and M. Pan, "Anomaly detection in smart grid based on encoder-decoder framework with recurrent neural network," *J. China Univ. Posts Telecommun.*, vol. 24, no. 6, pp. 67–73, 2017.

[77] P. Baraldi, F. Di Maio, D. Genini, and E. Zio, "Comparison of data-driven reconstruction methods for fault detection," *IEEE Trans. Reliab.*, vol. 64, no. 3, pp. 852–860, Sep. 2015.

[78] G. Kang, S. Gao, L. Yu, and D. Zhang, "Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2679–2690, Aug. 2019.

[79] H. Zhao, H. Liu, W. Hu, and X. Yan, "Anomaly detection and fault analysis of wind turbine components based on deep learning network," *Renew. Energy*, vol. 127, pp. 825–834, Nov. 2018.

[80] A. Shaheryar, X.-C. Yin, H.-W. Hao, H. Ali, and K. Iqbal, "A denoising based autoassociative model for robust sensor monitoring in nuclear power plants," *Sci. Technol. Nucl. Install.*, vol. 2016, pp. 1–17, Jan. 2016.

[81] H. Wang, H. Wang, G. Jiang, J. li, and Y. Wang, "Early fault detection of wind turbines based on operational condition clustering and optimized deep belief network modeling," *Energies*, vol. 12, no. 6, p. 984, 2019.

[82] L. Wang, Z. Zhang, H. Long, J. Xu, and R. Liu, "Wind turbine gearbox failure identification with deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1360–1368, Jun. 2017.

[83] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, Apr. 2019.

[84] F. Zhou, Y. Gao, and C. Wen, "A novel multimode fault classification method based on deep learning," *J. Control Sci. Eng.*, vol. 2017, Mar. 2017, Art. no. 3583610.

[85] S.-Y. Shao, W.-J. Sun, R.-Q. Yan, P. Wang, and R. X. Gao, "A deep learning approach for fault diagnosis of induction motors in manufacturing," *Chin. J. Mech. Eng.*, vol. 30, no. 6, pp. 1347–1356, 2017.

[86] Y. Liu, L. Duan, Z. Yuan, N. Wang, and J. Zhao, "An intelligent fault diagnosis method for reciprocating compressors based on LMD and SDAE," *Sensors*, vol. 19, no. 5, p. 1041, 2019.

[87] L. Jing, M. Zhao, P. Li, and X. Xu, "A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox," *Measurement*, vol. 111, pp. 1–10, Dec. 2017.

[88] F. Jia, Y. G. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 303–315, May 2016.

[89] M. Ma, C. Sun, and X. Chen, "Deep coupling autoencoder for fault diagnosis with multimodal sensory data," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1137–1145, Mar. 2018.

[90] X. Zhao, J. Wu, Y. Zhang, Y. Shi, and L. Wang, "Fault diagnosis of motor in frequency domain signal by stacked de-noising auto-encoder," *Comput. Mater. Continua*, vol. 57, no. 2, pp. 223–242, 2018.

[91] T. Wang, Y. He, B. Li, and T. Shi, "Transformer fault diagnosis using self-powered RFID sensor and deep learning approach," *IEEE Sensors J.*, vol. 18, no. 15, pp. 6399–6411, Aug. 2018.

[92] S. Dong, G. Wen, Z. Zhang, Y. Yuan, and J. Luo, "Rolling bearing incipient degradation monitoring and performance assessment based on signal component tracking," *IEEE Access*, vol. 7, pp. 45983–45993, 2019.

[93] H. Liu, L. Li, and J. Ma, "Rolling bearing fault diagnosis based on STFT-deep learning and sound signals," *Shock Vib.*, vol. 2016, Jul. 2016, Art. no. 6127479.

[94] L. H. Wang, X. P. Zhao, J. X. Wu, Y. Y. Xie, and Y. H. Zhang, "Motor fault diagnosis based on short-time Fourier transform and convolutional neural network," *Chin. J. Mech. Eng.*, vol. 30, pp. 1357–1368, Nov. 2017.

[95] P. Ma, H. Zhang, W. Fan, C. Wang, G. Wen, and X. Zhang, "A novel bearing fault diagnosis method based on 2D image representation and transfer learning-convolutional neural network," *Meas. Sci. Technol.*, vol. 30, no. 5, pp. 1–23, 2019.

[96] S. Guo, T. Yang, W. Gao, C. Zhang, and Y. Zhang, "An intelligent fault diagnosis method for bearings with variable rotating speed based on pythagorean spatial pyramid pooling CNN," *Sensors*, vol. 18, no. 11, p. E3857, 2018.

[97] M. H. Zhao, M. Kang, B. Tang, and M. Pecht, "Multiple wavelet coefficients fusion in deep residual networks for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4696–4706, Jun. 2019.

[98] S. Ma, W. Liu, W. Cai, Z. Shang, and G. Liu, "Lightweight deep residual CNN for fault diagnosis of rotating machinery based on depthwise separable convolutions," *IEEE Access*, vol. 7, pp. 57023–57036, 2019.

[99] W. Jiang, J. Zhou, H. Liu, and Y. Shan, "A multi-step progressive fault diagnosis method for rolling element bearing based on energy entropy theory and hybrid ensemble auto-encoder," *ISA Trans.*, vol. 87, pp. 235–250, Apr. 2019.

[100] V. H. Nguyen, J. S. Cheng, Y. Yu, and V. T. Thai, "An architecture of deep learning network based on ensemble empirical mode decomposition in precise identification of bearing vibration signal," *J. Mech. Sci. Technol.*, vol. 33, no. 1, pp. 41–50, 2019.

[101] F. Example, "Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings," *Shock Vib.*, vol. 2017, pp. 1–17 Oct. 2017.

[102] M. Sohaib, C.-H. Kim, and J.-M. Kim, "A hybrid feature model and deep-learning-based bearing fault diagnosis," *Sensors*, vol. 17, no. 12, p. 2876, 2017.

[103] J. Yu, "A selective deep stacked denoising autoencoders ensemble with negative correlation learning for gearbox fault diagnosis," *Comput. Ind.*, vol. 108, pp. 62–72, Jun. 2019.

[104] R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani, M. Saif, S. H. Kia, and H. Henao, and G.-A. Capolino, "Information fusion and semi-supervised deep learning scheme for diagnosing gear faults in induction machine systems," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6331–6342, Aug. 2019.

[105] C. Li, R.-V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera, "Fault diagnosis for rotating machinery using vibration measurement deep statistical feature learning," *Sensors*, vol. 16, no. 895, pp. 1–19, 2016.

[106] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

[107] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 19, no. 5, p. 1088, 2019.

[108] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," *Neural Comput. Appl.*, pp. 1–14, Feb. 2019.

[109] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

[110] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, "Online fault diagnosis method based on transfer convolutional neural networks," *IEEE Trans. Instrum. Meas.*, to be published.

[111] D. T. Hoang and H. J. Kang, "Rolling element bearing fault diagnosis using convolutional neural network and vibration image," *Cogn. Syst. Res.*, vol. 53, pp. 42–50, Jan. 2019.

[112] C. Lu, Z. Wang, and B. Zhou, "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification," *Adv. Eng. Inform.*, vol. 32, pp. 139–151, Apr. 2017.

[113] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, "Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3539–3549, Apr. 2018.

[114] X. Zhu, D. Hou, P. Zhou, Z. Han, Y. Yuan, W. Zhou, and Q. Yin, "Rotor fault diagnosis using a convolutional neural network with symmetrized dot pattern images," *Measurement*, vol. 138, pp. 526–535, May 2019.

[115] X. Zhu, J. Zhao, D. Hou, and Z. Han, "An SDP characteristic information fusion-based CNN vibration fault diagnosis method," *Shock Vib.*, vol. 2019, pp. 1–14, Feb. 2019.

[116] J. Li, X. Li, D. He, and Y. Qu, "A novel method for early gear pitting fault diagnosis using stacked SAE and GBRBM," *Sensors*, vol. 19, no. 4, p. 758, 2019.

[117] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network," *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2727–2736, Aug. 2018.

[118] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 185–195, Jan. 2018.

[119] L. Yu, J. Qu, F. Gao, and Y. Tian, "A novel hierarchical algorithm for bearing fault diagnosis based on stacked LSTM," *Shock Vib.*, vol. 2019, pp. 1–10, Oct. 2019.

[120] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks," *Renew. Energy*, vol. 133, pp. 422–432, Apr. 2018.

[121] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mech. Syst. Signal Process.*, vol. 110, pp. 349–367, Sep. 2018.

[122] C. Wu, P. Jiang, C. Ding, F. Feng, and T. Chen, "Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network," *Comput. Ind.*, vol. 108, pp. 53–61, Jun. 2019.

[123] D. Peng, Z. Liu, H. Wang, Y. Qin, and L. Jia, "A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains," *IEEE Access*, vol. 7, pp. 10278–12093, 2019.

[124] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.

[125] Y. Han, B. Tang, and L. Deng, "An enhanced convolutional neural network with enlarged receptive fields for fault diagnosis of planetary gearboxes," *Comput. Ind.*, vol. 107, pp. 50–58, May 2019.

[126] H. Pan, X. He, S. Tang, and F. Meng, "An improved bearing fault diagnosis method using one-dimensional CNN and LSTM," *J. Mech. Eng.*, vol. 64, nos. 7–8, pp. 443–452, May 2018.

[127] H. Li, J. Huang, and S. Ji, "Bearing fault diagnosis with a feature fusion method based on an ensemble convolutional neural network and deep neural network," *Sensors*, vol. 19, no. 9, p. E2034, 2019.

[128] X. Li, J. Li, Y. Qu, and D. He, "Gear pitting fault diagnosis using integrated CNN and GRU network with both vibration and acoustic emission signals," *Appl. Sci.*, vol. 9, no. 4, p. 768, Feb. 2019.

[129] Z. He, H. Shao, X. Zhang, J. Cheng, and Y. Yang, "Improved deep transfer auto-encoder for fault diagnosis of gearbox under variable working conditions with small training samples," *IEEE Access*, vol. 7, pp. 115368–115377, 2019.

[130] F. Jia and Y. Lei, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619–628, Jan. 2018.

[131] J. Wang, "Construction of a batch-normalized autoencoder network and its application in mechanical intelligent fault diagnosis," *Meas. Sci. Technol.*, vol. 30, no. 1, p. 015106, 2019.

[132] *The Case Western Reserve University Bearing Data Center Website*. Accessed: Aug. 13, 2019. [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home

[133] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 101–110, Feb. 2018.

[134] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 1, pp. 136–144, Jan. 2017.

[135] S. Wang, J. Xiang, Y. Zhong, and Y. Zhou, "Convolutional neural network-based hidden Markov models for rolling element bearing fault identification," *Knowl.-Based Syst.*, vol. 144, pp. 65–76, Mar. 2018.

[136] H. Jiang, X. Li, H. Shao, and K. Zhao, "Intelligent fault diagnosis of rolling bearings using an improved deep recurrent neural network," *Meas. Sci. Technol.*, vol. 29, no. 6, p. 065107, 2018.

[137] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke, "Deep learning for infrared thermal image based machine health monitoring," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 151–159, Jul. 2017.

[138] X. Xu, H. Zheng, Z. Guo, X. Wu, and Z. Zheng, "SDD-CNN: Small data-driven convolution neural networks for subtle roller defect inspection," *Appl. Sci.*, vol. 9, no. 7, p. 1364, 2019.

[139] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.*, vol. 8, no. 9, p. 1575, 2018.

[140] Z. Wang, C. Song, and T. Chen, "Deep learning based monitoring of furnace combustion state and measurement of heat release rate," *Energy*, vol. 131, pp. 106–112, Jul. 2017.

[141] W. Hou, Y. Wei, Y. Jin, and C. Zhu, "Deep features based on a DCNN model for classifying imbalanced weld flaw types," *Measurement*, vol. 131, pp. 482–489, Jan. 2019.

[142] P. Zhou, G. Zhou, Z. Zhu, C. Tang, Z. He, W. Li, and F. Jiang, "Health monitoring for balancing tail ropes of a hoisting system using a convolutional neural network," *Appl. Sci.*, vol. 8, no. 8, p. 1346, 2018.

[143] Z. Jia, Z. Liu, C.-M. Vong, and M. Pecht, "A rotating machinery fault diagnosis method based on feature learning of thermal images," *IEEE Access*, vol. 7, pp. 12348–12359, 2019.

[144] X. Li, Q. Yang, Z. Lou, and W. Yan, "Deep learning based module defect analysis for large-scale photovoltaic farms," *IEEE Trans. Energy Convers.*, vol. 34, no. 1, pp. 520–529, Mar. 2019.

[145] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017.

[146] D. Guo, M. Zhong, H. Ji, Y. Liu, and R. Yang, "A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors," *Neurocomputing*, vol. 319, pp. 155–163, Nov. 2018.

[147] J. Qiu, W. Liang, L. Zhang, X. Yu, and M. Zhang, "The early-warning model of equipment chain in gas pipeline based on DNN-HMM," *J. Natural Gas Sci. Eng.*, vol. 27, pp. 1710–1722, Nov. 2015.

[148] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4788–4797, Jun. 2019.

[149] Z. Wu, Y. Guo, W. Lin, S. Yu, and Y. Ji, "A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems," *Sensors*, vol. 18, no. 4, p. 1096, 2018.

[150] A. Y. Appiah, X. Zhang, B. B. K. Ayawli, and F. Kyeremeh, "Long short-term memory networks based automatic feature extraction for photovoltaic array fault diagnosis," *IEEE Access*, vol. 7, pp. 30089–30101, 2019.

[151] T. de Bruin, K. Verbert, and R. Babuska, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 523–533, Mar. 2017.

[152] H. Zhao, S. Sun, and B. Jin, "Sequential fault diagnosis based on LSTM neural network," *IEEE Access*, vol. 6, pp. 12929–12939, 2018.

[153] T. Shi, Y. He, T. Wang, and B. Li, "Open switch fault diagnosis method for PWM voltage source rectifier based on deep learning approach," *IEEE Access*, vol. 7, pp. 66595–66608, 2019.

[154] C. Zhang, Y. He, L. Yuan, and S. Xiang, "Analog circuit incipient fault diagnosis method using DBN based features extraction," *IEEE Access*, vol. 6, pp. 23053–23064, 2018.

[155] Y. Guo, Z. Tan, H. Chen, G. Li, J. Wang, R. Huang, J. Liu, and T. Ahmad, "Deep learning-based fault diagnosis of variable refrigerant flow air-conditioning system for building energy saving," *Appl. Energy*, vol. 225, pp. 732–745, 2018.

[156] X. Qin, Y. Zhang, W. Mei, G. Dong, J. Gao, P. Wang, J. Deng, and H. Pan, "A cable fault recognition method based on a deep belief network," *Comput. Elect. Eng.*, vol. 71, vol. 71, pp. 452–464, Aug. 2018.

[157] D. Yu, Z. M. Chen, K. S. Xiahou, M. S. Li, T. Y. Ji, and Q. H. Wu, "A radically data-driven method for fault detection and diagnosis in wind turbines," *Int. J. Electr. Power Energy Syst.*, vol. 99, pp. 577–584, Jul. 2018.

[158] D. Jiejie, S. Hui, and S. Gehao, "Dissolved gas analysis of insulating oil for power transformer fault diagnosis with deep belief network," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 24, no. 5, pp. 2828–2835, Oct. 2017.

[159] J. Yin and W. Zhao, "Fault diagnosis network design for vehicle on-board equipments of high-speed railway: A deep learning approach," *Eng. Appl. Artif. Intell.*, vol. 56, pp. 250–259, Nov. 2016.

[160] J. Ma, S. Ni, W. Xie, and W. Dong, "Deep auto-encoder observer multiple-model fast aircraft actuator fault diagnosis algorithm," *Int. J. Control Autom. Syst.*, vol. 15, no. 4, pp. 1641–1650, Aug. 2017.

[161] J. Yu, X. Zheng, and S. Wang, "Stacked denoising autoencoder-based feature learning for out-of-control source recognition in multivariate manufacturing process," *Qual. Reliab. Eng. Int.*, vol. 35, no. 1, pp. 204–223, 2019.

[162] Z. Zhang, S. Li, Y. Xiao, and Y. Yang, "Intelligent simultaneous fault diagnosis for solid oxide fuel cell system based on deep learning," *Appl. Energy*, vols. 233–234, pp. 930–942, Jan. 2019.

[163] Y. Wang, M. Liu, Z. Bao, and S. Zhang, "Stacked sparse autoencoder with PCA and SVM for data-based line trip fault diagnosis in power systems," *Neural Comput. Appl.*, vol. 5, pp. 1–13, Apr. 2018.

[164] G. Hu, H. Li, Y. Xia, and L. Luo, "A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis," *Comput. Ind.*, vol. 100, pp. 287–296, Sep. 2018.

[165] Q. Tang, Y. Chai, J. Qu, and H. Ren, "Fisher discriminative sparse representation based on DBN for fault diagnosis of complex system," *Appl. Sci.*, vol. 8, no. 5, p. 795, 2018.

[166] Z. Gao, C. Ma, D. Song, and Y. Liu, "Deep quantum inspired neural network with application to aircraft fuel system fault diagnosis," *Neurocomputing*, vol. 238, pp. 13–23, May 2017.

[167] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0," *Inf. Fusion*, vol. 50, pp. 92–111, Oct. 2019.

[168] D. Chen, S. Yang, and F. Zhou, "Transfer learning based fault diagnosis with missing data due to multi-rate sampling," *Sensors*, vol. 19, no. 8, p. 1826, Apr. 2019.

[169] S. Wang, S. Fan, J. Chen, X. Liu, B. Hao, and J. Yu, "Deep-learning based fault diagnosis using computer-visualised power flow," *IET Gener. Transm. Distrib.*, vol. 12, no. 17, pp. 3985–3992, 2018.

[170] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, "Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2416–2425, Apr. 2019.

[171] P. Khumprom and N. Yodo, "A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm," *Energies*, vol. 12, no. 4, p. 660, Feb. 2019.

[172] L. Ren, J. Cui, Y. Sun, and X. Cheng, "Multi-bearing remaining useful life collaborative prediction: A deep learning approach," *J. Manuf. Syst.*, vol. 43, pp. 248–256, Apr. 2017.

[173] J. Deutsch, M. He, and D. He, "Remaining useful life prediction of hybrid ceramic bearings using an integrated deep learning and particle filter approach," *Appl. Sci.*, vol. 7, no. 7, p. 649, 2017.

[174] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 1, pp. 11–20, Jan. 2018.

[175] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proc. Int. Conf. Prognostics Health Manage.*, 2008, pp. 1–6.

[176] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for remaining useful life estimation," *Comput. Ind.*, vol. 108, pp. 186–196, Jun. 2019.

[177] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *J. Manuf. Syst.*, vol. 48, pp. 78–86, Jul. 2018.

[178] C. Zhang, P. Lim, A. K. Qin, S. Member, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2016.

[179] L. Wen, Y. Dong, and L. Gao, "A new ensemble residual convolutional neural network for remaining useful life estimation," *Math. Biosci. Eng.*, vol. 16, no. 2, pp. 862–880, 2019.

[180] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliab. Eng. Syst. Saf.*, vol. 182, pp. 208–218, Feb. 2019.

[181] Y. T. Wu, M. Yuan, S. Dong, L. Li, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.

[182] A. L. Ellefsen, S. Ushakov, V. Æsøy, and H. Zhang, "Validation of data-driven labeling approaches using a novel deep network structure for remaining useful life predictions," *IEEE Access*, vol. 7, pp. 71563–71575, 2019.

[183] W. Mao, J. He, J. Tang, and Y. Li, "Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network," *Adv. Mech. Eng.*, vol. 10, no. 12, pp. 1–18, 2018.

[184] Z. Rui, Y. Ruqiang, W. Jinjiang, and M. Kezhi, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, pp. 1–18, 2017.

[185] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1539–1548, Feb. 2018.

[186] F. Aghazadeh, A. Tahan, and M. Thomas, "Tool condition monitoring using spectral subtraction algorithm and artificial intelligence methods in milling process," *Int. J. Mech. Eng. Robot. Res.*, vol. 6, no. 6, pp. 30–34, 2018.

[187] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017.

[188] Q. Wang, B. Zhao, H. Ma, J. Chang, and G. Mao, "A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional convolutional neural network with signal conversion," *J. Mech. Sci. Technol.*, vol. 33, no. 6, pp. 2561–2571, Jun. 2019.

[189] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, "Metrics for evaluating performance of prognostic techniques," in *Proc. Int. Conf. Prognostics Health Manage. (PHM)*, 2008, pp. 1–17.

[190] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliab. Eng. Syst. Saf.*, vol. 183, no. Jun. 2018, pp. 240–251, 2019.

[191] A. Zhang, H. Wang, S. Li, Y. Cui, Z. Liu, G. Yang, and J. Hu, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Appl. Sci.*, vol. 8, no. 12, p. 2416, Nov. 2018.

[192] J. Yoo and J. G. Baek, "A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network," *Appl. Sci.*, vol. 8, no. 7, p. 1102, 2018.

[193] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7076–7083, Nov. 2016.

[194] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3208–3216, Apr. 2019.

[195] A. Saxena and K. Goebel, *Turbofan Engine Degradation Simulation Data Set*. Moffett Field, CA, USA: NASA Ames Research Center, 2008. Accessed: Aug. 13, 2019. [Online]. Available: http://ti.arc.nasa.gov/project/prognostic-data-repository

[196] P. Nectoux, "PRONOSTIA: An experimental platform for bearings accelerated degradation tests," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (PHM)*, 2012, pp. 1–8.

[197] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.

[198] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, and L. Zhang, "Remaining useful life prediction for lithium-ion battery: A deep learning approach," *IEEE Access*, vol. 6, pp. 50587–50598, 2018.

[199] G.-W. You, S. Park, and D. Oh, "Diagnosis of electric vehicle batteries using recurrent neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4885–4893, Jun. 2017.

[200] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5695–5705, Jul. 2018.

[201] J. Ma, H. Su, W. Zhao, and B. Liu, "Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning," *Complex*, vol. 2018, pp. 1–13, Jul. 2018.

[202] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.

[203] L. Ren, Y. Sun, H. Wang, and L. Zhang, "Prediction of bearing remaining useful life with deep convolution neural network," *IEEE Access*, vol. 6, pp. 13041–13049, 2018.

[204] L. Ren, X. Cheng, X. Wang, J. Cui, and L. Zhang, "Multi-scale dense gate recurrent unit networks for bearing remaining useful life prediction," *Future Gener. Comput. Syst.*, vol. 94, pp. 601–609, May 2019.

[205] L. Ren, Y. Sun, J. Cui, and L. Zhang, "Bearing remaining useful life prediction based on deep autoencoder and deep neural networks," *J. Manuf. Syst.*, vol. 48, pp. 71–77, Jul. 2018.

[206] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[207] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[208] S. Shao, P. Wang, and R. Yan, "Generative adversarial networks for data augmentation in machine fault diagnosis," *Comput. Ind.*, vol. 106, pp. 85–93, Apr. 2019.

[209] H. Liu, J. Zhou, Y. Xu, Y. Zheng, X. Peng, and W. Jiang, "Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks," *Neurocomputing*, vol. 315, pp. 412–424, Nov. 2018.

[210] J. Liu, F. Qu, X. Hong, and H. Zhang, "A small-sample wind turbine fault detection method with synthetic fault data using generative adversarial nets," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 3877–3888, Jul. 2019.

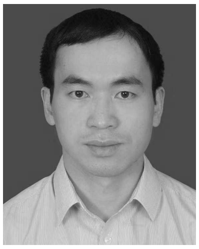
[211] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5525–5534, Jul. 2019.

[212] W. Mao, Y. Liu, L. Ding, and Y. Li, "Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study," *IEEE Access*, vol. 7, pp. 9515–9530, 2019.

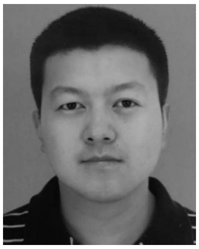
LIANGWEI ZHANG received the Ph.D. degree in operation and maintenance engineering from the Luleå University of Technology, Luleå, Sweden, in 2017. He is currently a Lecturer with the Department of Industrial Engineering, Dongguan University of Technology, Dongguan, China. He is also an Adjunct Lecturer with the Division of Operation and Maintenance Engineering, Luleå University of Technology. His research interests include machine learning, fault detection, and prognostics and health management.

JING LIN received the Ph.D. degree in management with the Nanjing University of Science and Technology, in April 2008. She is currently an Associate Professor with the Division of Operation and Maintenance Engineering, Luleå University of Technology (LTU), Luleå, Sweden. She is also a Joint Professor with Beijing Jiaotong University, Beijing, China. Her research interests include Bayesian reliability modeling, maintenance engineering, and prognostics and health management.

BIN LIU received the B.S. degree in automation from Zhejiang University, China, and the Ph.D. degree in industrial engineering from the City University of Hong Kong, Hong Kong. He was a Postdoctoral Fellow with the University of Waterloo, Canada. He is currently a Lecturer with the Department of Management Science, University of Strathclyde, Glasgow, U.K. His research interests include risk analysis, reliability and maintenance modeling, decision making under uncertainty, and data analysis.



ZHICONG ZHANG received the B.E. degree from the Department of Mechanical Engineering and the Ph.D. degree from the Department of Industrial Engineering, Tsinghua University, Beijing, China, in 2002 and 2007, respectively. He is currently a Professor with the Department of Industrial Engineering, Dongguan University of Technology, Guangdong, China. He has authored/coauthored over 30 publications and one book. His research interests are industrial engineering, scheduling, operations research, and machine learning.



XIAOHUI YAN received the B.S. degree in industrial engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in mechatronic engineering from the University of Chinese Academy of Sciences, in 2013. He is currently an Associate Professor with the Dongguan University of Technology. His current research interests include swarm intelligence, bio-heuristic computation, neural networks, and their applications.

MUHENG WEI received the Ph.D. degree from Tsinghua University, in 2013. She is currently a Senior Engineer with the Oceanic Intelligent Technology Innovation Center, CSSC Systems Engineering Research Institute, Beijing, China. Her research interests include cyber-physical systems, industrial AI, industrial big data, and prognostic and health management.