IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 7, 2019, accepted April 29, 2019, date of publication May 3, 2019, date of current version May 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914731

A Multi-Level-Denoising Autoencoder Approach for Wind Turbine Fault Detection

XIN WU[1], GUOQIAN JIANG[1], (Member, IEEE), XIAO WANG[1], PING XIE[1], AND XIAOLI LI[2]

[1]School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
[2]National Key Laboratory of Cognitive Neuroscience and Learning, Beijing Normal University, Beijing 100875, China

Corresponding authors: Guoqian Jiang (jiangguoqian@ysu.edu.cn) and Ping Xie (pingx@ysu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61803329 and Grant 51575472, in part by the Natural Science Foundation of Hebei Province of China under Grant F2018203413 and Grant F2016203421, in part by the China Postdoctoral Science Foundation under Grant 2018M640247, in part by the Key Research and Development Program of Qinhuangdao under Grant 201805A005, and in part by the Doctoral Foundation of Yanshan University under Grant BL18040.

ABSTRACT The effective fault detection of wind turbines (WTs) can greatly help to improve their availability and reduce their operation and maintenance costs. In this context, data-driven fault detection approaches have attracted a lot of interests due to the availability of a large amount of monitoring sensor data containing rich information related to health conditions of WTs. However, sensor data collected from WTs are naturally multivariate and highly nonlinear correlated with redundant information and significantly contaminated measurement noise, which makes the WT fault detection more challenging. To this end, this paper develops a multivariate data-driven fault detection (MDFD) framework based on a recently emerged neural network algorithm named denoising autoencoder (DAE). Instead of using a single fixed noise level in the traditional DAE, a novel multi-level-denoising autoencoder (MLD-AE) method is proposed to enhance the representation learning ability by designing different multi-level noise adding schemes. The proposed MLD-AE could better discover useful patterns at multiple corrupted scales and capture nonlinear dependencies from noisy multivariate sensor data, therefore robustly reconstruct the original signal with the preserved largest information. The proposed framework and method are evaluated on both simulated data from a generic 5 MW WT benchmark and SCADA data from a real wind farm. The results demonstrate that our proposed MLD-AE-based fault detection approach significantly outperforms traditional DAE, AE, and linear PCA approaches, which has great potentials for practical applications in the wind industry.

INDEX TERMS Wind turbines (WTs), fault detection, multivariate data-driven, denoising autoencoder (DAE), multi-level-denoising autoencoder (MLD-AE).

I. INTRODUCTION

Nowadays, wind energy is experiencing a rapid growth among various renewable energy sources worldwide, and accordingly, wind turbines (WTs) have been widely stalled onshore and offshore. Due to harsh working environments and constantly changing loads and speeds, WTs are subject to various faults/failures in sensors, actuators, and components, thus resulting in unscheduled downtime and high maintenance and operation O&M costs [1], [2]. Statistically, the WT O&M costs account for 20-25% of the overall cost of power generation. In order to reduce O&M costs and maximize the uptime of WTs, it is highly desirable to develop advanced fault detection systems to detect those possible faults as early as possible. Early detection of failures can assist operators in wind farms to take proper actions in time to avoid secondary damage and even catastrophic accidents, thus enabling better maintenance planning and logistics.

In recent years, considerable efforts have been devoted to the problem of condition monitoring and fault detection of WTs and their subsystems or key components, including sensors [3], bearings [4], [5], gearboxes [6], [7], and pitch actuators [8], etc. Among them, vibration and lubrication oil monitoring are two widely used ones. To effectively and reliably detect mechanical faults from the complex raw sensory signal, such as vibration signals, different signal processing algorithms have been developed to extract fault signatures [4], [9], [10] and finally identify and diagnose faults. However, both techniques are sophisticated and expensive due to the need of additional installation of sensors and data acquisition hardware devices; furthermore, the interpretation of the analysis results also require high-level

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

knowledge about the monitored component. Alternatively, Supervisory Control and Data Acquisition (SCADA) monitoring has been considered as a cost-effective means [11] for early warning of failures and performance issues since no additional sensors and hardware devices are required, and recently, it has attracted significant interest. Practically, modern large-scale WTs have been equipped with a SCADA system to record and collect the operation data and the status data. Typically, these data include meteorological conditions (e.g. wind speed, wind direction), bearing temperatures, pressures, and power and electrical measurements [12], which contain rich information concerning the health of the WT and its key components or subsystems. The availability of a large amount of historical operational data facilitates the wide application of data-driven fault detection methods.

Practically, the operational condition of the turbine is monitored mainly based on simple limit sensing. The method determines the thresholds for each observation but ignores the correlations among multiple measurements. Hence, it often leads to false alarms or missed detections of faults. Indeed, SCADA data are multivariate in nature and highly correlated due to the interaction and independence between different subsystems in a WT. To address such an issue, much effort has been expended to investigate the correlations among the SCADA variables to develop efficient and reliable monitoring techniques. One effective solution of the use of SCADA data for condition monitoring is normal behavior modeling. In literature, various machine learning and data mining algorithms, like neural networks (NN) [13]–[15], support vector regression (SVR) [16], random forest (RF) [17], long short-term memory network (LSTM) [18] and recently emerged deep neural networks (DNN) [19], have been extensively used to develop normal behavior models. Then, differences between the estimated behavior and actual observed behavior are used to identify the presence of potential faults. These methods mainly focus on predicting a certain output variable (e.g. gearbox bearing temperature, gearbox cooling oil temperature, and the generator temperature), given one or more input variables. Indeed, the health of a component or subsystem will be related to multiple variables.

On the other hand, in order to deal with correlations hidden in multiple sensor variables, multivariate static monitoring approaches like principal component analysis (PCA) and fisher discriminant analysis (FDA) have been applied for WT monitoring [20], [21]. However, multivariate sensor data collected from WTs are highly nonlinear correlated. Traditional multivariate monitoring approaches like PCA and FDA based on the assumption of linear process behavior may not be valid in practical situations. Alternatively, autoencoder (AE), as a powerful nonlinear modeling approach, has been proposed for fault detection purpose. The AE is a special feedforward neural network [22], which reproduces its input at the output layer. It has been successfully applied to missing data recovery [23], fault diagnosis in rotary machines [24], gas turbines [25], and wind turbine blades [26]. A notable feature of the AE approach is that it can learn arbitrary relationships among different sensor variables in both linear and nonlinear cases and therefore be more flexible than existing linear monitoring methods. Notably, WTs are driven stochastically by the wind and subject to various external disturbances, thus resulting in more noise and variations of data, which will in turn increase the difficulty to discover abnormal pattern from noisy multivariate data. However, the existing PCA-based and AE-based methods are often sensitive to the disturbances and noise, thus leading to biased monitoring results. Recent studies in the deep learning community have shown that denoising autoencoder (DAE), a variant of the traditional AE, can learn more robust representation and has improved generalization capability [27], especially for noisy input data. This good property of DAE is well suitable for dealing with noisy multivariate sensor data measured from WTs operating in variable conditions [28]. This motivates us to develop a new DAE-based fault detection system for WTs. It is well known that DAE adopts a denoising training scheme to produce the learned representations robust to partial corruption of the input pattern. However, during the training process of DAE, only a single fixed noise level is used and keeps unchanged, which can only capture partial information at a single corrupted scale. In order to improve representation learning ability of the traditional DAE, instead of using a single noise level, a new multi-level noise adding scheme is proposed to train the DAE model, with the aim to capture more meaningful nonlinear representation from multivariate noisy data and obtain robust signal reconstruction.

The contribution of this paper is two-fold. First, we propose a new multi-level-denoising autoencoder (MLD-AE) algorithm with two different multi-level noise adding schemes to train the normal behavior model under a general multivariate data-driven fault detection (MDFD) framework. Compared with the traditional DAE, the proposed MLD-AE will discover more useful patterns and nonlinear dependencies among multivariate noisy data, therefore could obtain robust signal reconstruction for further fault detection purpose. Second, we comprehensively evaluate the proposed algorithm using a generic simulated benchmark model and practical field measured SCADA data from a real wind farm. Specifically, we investigate the effects of several important parameters in MLD-AE algorithms with two commonly used noise types on detection performance, and conduct the comparative study with other traditional approaches.

The rest of this paper is organized as follows. A new MLD-AE approach is proposed in Section II. Then Section III presents a generic multivariate data-driven fault detection (MDFD) framework for WT systems. Performance evaluation is carried out on simulated benchmark model and real SCADA data in Section IV and Section V, respectively. Finally, conclusions are drawn in Section VI.

II. MULTI-LEVEL-DENOISING AUTOENCODER

In this section, we propose a new MLD-AE algorithm to learn robust representation from multivariate sensor data while capture nonlinear correlations hidden among different

sensor variables and result in more robust reconstruction, thus enabling to perform the subsequent detection task. We start by briefly describing the conventional DAE and point out its limitations for data representation learning. Then, we present our MLD-AE approach in detail.

A. BASIC DAE

DAE is a variation of conventional AE, which is trained to reconstruct a data sample $x \in \mathbb{R}^m$ from its corrupted version $\tilde{x}$ [27]. By doing so, it can prevent AE from just simply learning an identity mapping between the input x and the reconstructed output $\hat{x}$, and therefore learn more robust and informative representation from noisy data. It should be noted that the corrupted input $\tilde{x}$ is typically drawn from a conditional distribution $p(\tilde{x}|x)$. Two common choices are additive Gaussian (GS) noise and Zero-Masking (ZM) noise where a fraction of input values are randomly forced to 0 [29]. The former one is a natural choice for real-valued inputs, while the latter one can be viewed as turning off components considered missing or replacing their value by a default value. Both choices will be considered and compared in this study.

Specifically, like the conventional AE, DAE also consists of an encoder process and a decoder process. The corrupted input $\tilde{x}$ is first mapped to a hidden representation $h \in \mathbb{R}^l$ as (1) by the nonlinear transformation between the input layer and the hidden layer:

$$h = f(W_1\tilde{x} + b) \tag{1}$$

where $W_1 \in \mathbb{R}^{l\times m}$ is the weight matrix and $b \in \mathbb{R}^l$ is the bias vector. In this study, we use the sigmoid function $f(x) = 1/(1 + \exp(-x))$ for the nonlinear mapping purpose. The hidden layer code h can be viewed as a compression of input data with some information loss when number of hidden units is less than the number of input units, and it can capture the main variations in multivariate input data and eliminate less important information through dimension reduction.

Then, the hidden representation h is mapped back to a reconstruction output $\hat{x}$ as (2) through the decoder:

$$\hat{x} = g(W_2h + c) \tag{2}$$

where $W_2 \in \mathbb{R}^{m\times l}$ is the weight matrix, $c \in \mathbb{R}^m$ is the bias vector, and $g(\cdot)$ is the sigmoid function. The training process of DAE is to find optimal parameters $\boldsymbol{\theta} = \{W_1, W_2, b, c\}$ by minimizing the squared reconstruction error of the cost function as follows:

$$L(x, \tilde{x}) = \sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 \tag{3}$$

where n is the number of training data.

B. PROPOSED MLD-AE

As described in Section II-A, DAE is to be able to reconstruct data from an input of corrupted data and force the hidden layer to learn only the more robust features. Traditionally, a given input data will be stochastically corrupted using a single predefined noise level. Also, the noise level is kept fixed during the whole training process of DAE. Recent stuides [6], [30] in the deep learning community with classification tasks have shown that It should be noted that different noise levels will affect the learned representations. Specifically, for a high noise level, the input data will be more heavily corrupted during training, and force the network to learn coarse-grained features; whereas for a low noise level, the inputs will be only slightly corrupted and thus the network will be forced to learn fine-grained features. Motivated by this, we attempt to design a new multi-level noise adding scheme to enhance the representation learning capacity of the traditional DAE. Instead of a single fixed noise level, multiple different noise levels are used to train the autoencoder network, thus enables to learn more general and detailed feature patterns simultaneously at different scales from the original input data.

In this study, we consider two different schemes to add multiple noise levels. The first scheme is to train a single DAE sequentially using decreasing noise levels, such that $\upsilon_0 > \cdots > \upsilon_T \geq 0$, where υ_0 and υ_T are the initial noise level and the final noise level, respectively. We call the first scheme **MLD-AE-S1**. In the second scheme, we first corrupt the original input data X at different noise levels $\{\upsilon_0, \cdots, \upsilon_T\}$ to obtain the corrupted data $\{\tilde{X}_{\upsilon_0}, \cdots, \tilde{X}_{\upsilon_T}\}$. Then we concatenate all corrupted inputs to obtain the augmented data $\tilde{X}_{new}$ as the new input to train a single DAE. The second scheme is called **MLD-AE-S2**. Both training schemes are summarized in **Algorithm 1** and **Algorithm 2**, respectively. As is mentioned before, we will also investigate the effects of two commonly used noise types, i.e., GS noise and ZM noise on detection performance in Section IV and Section V.

C. MANIFOLD INTERPRETATION OF MLD-AE

As described in [27], DAE can be viewed as a way to define and learn a manifold from the given data. Similarly, we attempt to give a manifold interpretation of MLD-AE. Different from DAE, in our proposed MLD-AE, the original training samples are corrupted by using multiple different noise levels, and thus will lead to different learned representations containing partial information at different scales. We take the Zero-Masking noise as an example for further explanation. As shown in Figure 1, for a high level noise, the model is forced to reconstruct the original data using only few input variables; hence it learns only about manifold around larger neighborhood of input, and captures the general features about the input data. For a low level noise, the model can reconstruct the original input more easily with more input variables, and looks at smaller neighborhood around manifold about the data to capture detailed features. Therefore, the model trained with multiple noise levels can learn and incorporate useful information at different scales about the input data, thus enable to learn the manifold in a better way than a single fixed noise level. Therefore, MLD-AE has the more powerful ability to capture nonlinear hidden

Algorithm 1 MLD-AE Training Scheme I

procedure MLD-AE-S1$(X, \upsilon_0, \Delta\upsilon, \upsilon_T, K)$
  $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{m\times n}$ is the input matrix, in which $x_i \in [0,1]^m (1 \le i \le m)$
  N is the number of epochs to be iterated
  b is the number of batches
  η is the learning rate
  υ_0 is the initial noise level
  υ_T is the final noise level, $\upsilon_T < \upsilon_0$
  $\Delta\upsilon$ is the noise decreasing step size
  K is the iteration steps on the noise level $\upsilon_t (0 < t < T)$
  $\boldsymbol{\theta} = \{W_1, W_2, b, c\}$ is the parameters of a DAE, where $W_1 \in \mathbb{R}^{l\times m}$, $W_2 \in \mathbb{R}^{m\times l}$, $b \in \mathbb{R}^l$, $c \in \mathbb{R}^m$
  # Train DAE using initial noise level υ_0
  for epoch $i = 1$ to N **do**
    for mini-batch 1 to b **do**
      $\tilde{X}_0 = \text{getCorrupted}(X, \upsilon_0)$ ▷ The corruption process can be GS or ZM
      $h = sigmoid(W_1 * \tilde{X} + b)$
      $\hat{X} = sigmoid(W_2 * h + c)$
      $L(X, \hat{X}) = \sum \|X - \hat{X}\|^2$
      $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \partial L / \partial \boldsymbol{\theta}$
    end for
  end for
  # Continue to train using decreasing noise levels
  for t in $1, \cdots, T$ **do**
    $\upsilon_t := \upsilon_{t-1} - \Delta\upsilon$
    for 1 to K **do**
      for mini-batch 1 to b **do**
        $\tilde{X} = \text{getCorrupted}(X, \upsilon_t)$
        Take steps (14)-(17)
      end for
    end for
  end for
end procedure

Algorithm 2 MLD-AE Training Scheme II

procedure MLD-AE-S2$(X, \upsilon_0, \Delta\upsilon, \upsilon_T)$
  $X, \upsilon_0, \upsilon_0, \Delta\upsilon$ are same as ones in **MLD-AE-S1**
  for epoch $i = 1$ to N **do**
    for mini-batch 1 to b **do**
      $\tilde{X}_0 = \text{getCorrupted}(X, \upsilon_0)$
      for $t = 1$ to T **do**
        $\upsilon_t := \upsilon_{t-1} - \Delta\upsilon$
        $\tilde{X}_t = \text{getCorrupted}(X, \upsilon_t)$
      end for
      $\tilde{X}_{new} = [\tilde{X}_0, \cdots, \tilde{X}_T] \in \mathbb{R}^{n(T+1)\times m}$ ▷ Concatenate all corrupted inputs on different noise levels
      $h = sigmoid(W_1 * \tilde{X}_{new} + b)$
      $\hat{X} = sigmoid(W_2 * h + c)$
      $L(X, \hat{X}) = \sum \|X - \hat{X}\|^2$
      $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \partial L / \partial \boldsymbol{\theta}$
    end for
  end for
end procedure

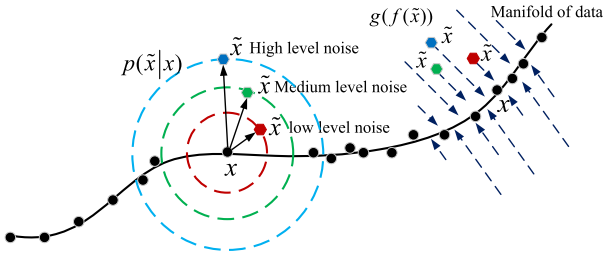

FIGURE 1. Manifold of data under different noise levels. Suppose training data x concentrate near a low-dimensional manifold (black line), and then the corrupted examples $\tilde{x}$ using different noise levels corresponding to different circles will lie farther from the manifold. These corrupted ones will be learned and projected back onto the manifold.

correlations from multivariate noisy sensor data, and will be used for building WT normal behavior models using SCADA data in the following section.

III. MULTIVARIATE DATA-DRIVEN FAULT DETECTION FRAMEWORK FOR WTs

The proposed generic MDFD framework for WTs is illustrated in Figure 2, where the MLD-AE approach described in Section II is used for normal behavior modeling using normal data collected from WTs. The main idea of this framework is based on the evaluation of the reconstruction error between the actual signal values and the reconstructed values from well-trained normal behavior model. The changes of the reconstruction error will give an indication of possible faults. Usually, normal test samples will produce a low reconstruction error since they can well satisfy the learned normal model; whereas the faulty samples will produce a high reconstruction error and therefore be identified as faults. As shown in Figure 2, the proposed MDFD framework mainly consists of two phases: *offline training phase* and *online detection phase*. The normal behavior model is built based on our proposed MLD-AE approach in the offline training phase, and the corresponding fault indicator and the detection threshold are calculated. During the online detection phase, the online fault indicator of a newly collected data sample is calculated and then compared with the detection threshold to determine whether a fault occurs. A fault alarm will be then triggered when the testing fault indicator exceeds a predefined threshold.

A. OFFLINE TRAINING PHASE

Let $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{m\times n}$ be the collected sensor data from a WT, where n denotes the number of data samples and m is the number of sensor variables. These variables are highly correlated due to their inherent nonlinear dynamics and interactions among multiple components of the WT

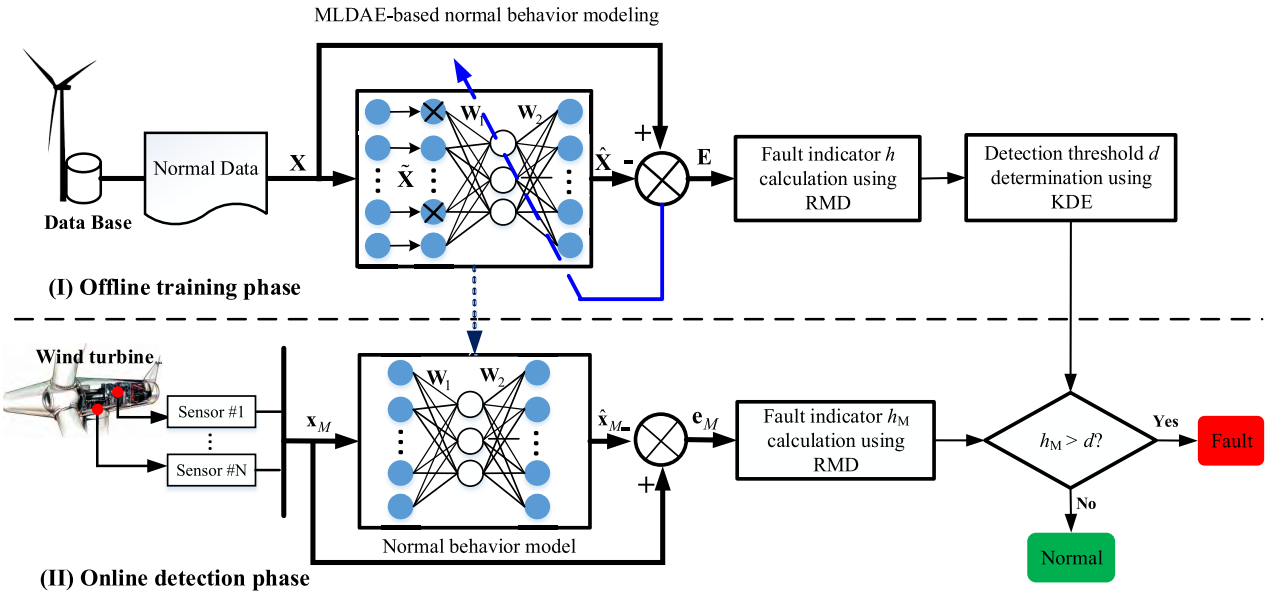

FIGURE 2. Designed multivariate data-driven fault detection framework for WTs with offline training phase and online detection phase. A normal behavior model based on the proposed MLD-AE is trained offline with corrupted normal data $\tilde{X}$, and then the learned model is used to calculate the residual e_M between a new coming data sample x_M and its corresponding reconstructed signal $\hat{x}_M$ for online fault detection.

system and more importantly, are redundant due to similar sensor locations.

1) NORMAL BEHAVIOR MODELING

In practice, normal data are much more easily obtained than faulty data due to WTs are mostly operating under normal conditions. Therefore, in this study, we focus on building normal behavior model using historical data under normal operating conditions of WTs. As is mentioned before, SCADA data are usually nonlinear correlated with significant measurement noises. In order to learn the relations among different sensor variables and capture data structure hidden in multivariate sensor data, in this paper, the proposed MLD-AE approach is adopted for data analysis and modeling. It should be noted that during the training phase, normal data are generally unlabeled, and therefore it can be considered as an unsupervised learning method. A major advantage of using normal behavior models is that no prior knowledge about the signal behavior is required. Specifically, MLD-AE-S1 and MLD-AE-S2 described in Section II-B are considered to learn two important weight matrices W_1 between the input layer and the hidden layer and W_2 connecting the hidden layer and the output layer from the corrupted input $\tilde{X}$ with multiple noise levels. Thus, the learned relations are embedded in learned weight metrics and bias vectors, which can be further used to perform the reconstruction of new data samples during online detection phase.

2) FAULT INDICATOR CALCULATION

Once the trained normal behavior model is trained, we can calculate the residuals E as the difference between the training inputs and the reconstructed outputs as follows:

$$E = \hat{X} - X \tag{4}$$

Then, the fault indicator h of kth sample is calculated by using Mahalanobis distance (MD) as follows:

$$h_k = \sqrt{(E_k - \boldsymbol{\mu})C^{-1}(E_k - \boldsymbol{\mu})^T} \tag{5}$$

where $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_m]$ is the vector of mean values and $\mathbf{C}$ is the covariance matrix both obtained from the training set. MD is an effective metric for multivariate problems with interaction effects among large numbers of variables, and can give a univariate distance value for the residual vector $\mathbf{E}$.

However, considering that both the sample mean and the sample covariance are sensible to the presence of outliers, in this study, we use robust mahalanobis distance (RMD) [31] [32] to calculate the fault indicator, which is defined as:

$$h_k = \sqrt{(E_k - \hat{\boldsymbol{\mu}})MCD^{-1}(E_k - \hat{\boldsymbol{\mu}})^T} \tag{6}$$

where $\hat{\boldsymbol{\mu}}$ is the robust measure of central tendency (the median) and MCD^{-1} is the inverse covariance matrix calculated from the sample population through the minimum covariance determinant estimator.

3) DETECTION THRESHOLD DETERMINATION

The threshold is determined by the distribution of the fault indicator h. Because h does not follow the Gaussian distribution, we use the kernel density estimation (KDE) [33] to calculate the threshold from the estimated probability density function.

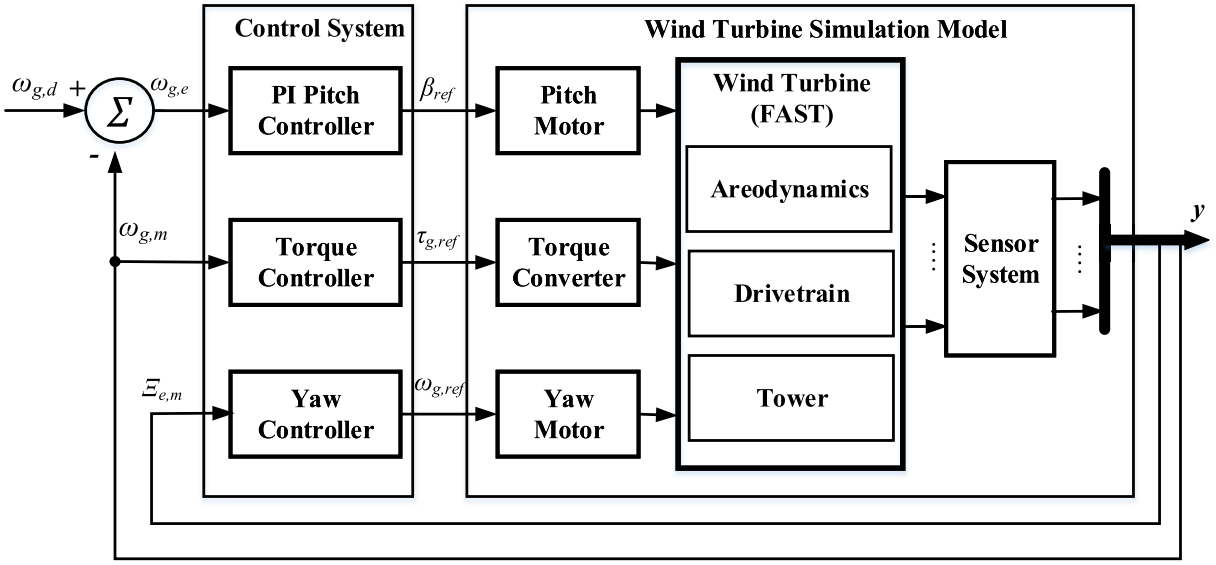

FIGURE 3. Block diagram of wind turbine benchmark model.

Assuming a set of data points $x_k, k = 1, 2, \cdots, N$, the KDE at point x is defined as:

$$p(x) = \frac{1}{N\sigma}\sum_{k=1}^{N} K(\frac{x - x_k}{\sigma}) \tag{7}$$

where $K(\cdot)$ is the kernel function, and σ is the bandwidth. In this paper, the Gaussian kernel presented in (8) is used.

$$K(g) = \frac{e^{-\frac{g^2}{2}}}{\sqrt{2\pi}} \tag{8}$$

Then, the detection threshold d with a given confidence level α is formulated as:

$$P(x < d) = \int_{-\infty}^{d} p(x)dx = \alpha \tag{9}$$

B. ONLINE DETECTION PHASE

During this phase, the new online monitoring samples will be tested to identify the faults based on the well trained normal behavior model and the predetermined threshold obtained in the offline training phase.

For a new incoming sample $x_M \in \mathbb{R}^m$, we first calculate the model estimated output $\hat{x}_M \in \mathbb{R}^m$ using the learned weight matrices $W_1 \in \mathbb{R}^{l\times m}$ and $W_2 \in \mathbb{R}^{m\times l}$ (l is the number of hidden units) obtained from offline training phase as:

$$\hat{x}_M = g(W_2 f(W_1 x_M + b) + c) \tag{10}$$

where $f(\cdot)$ and $g(\cdot)$ are activation function, $b \in \mathbb{R}^l$ and $c \in \mathbb{R}^m$ are learned bias vector.

And then the residual e_M between the actual measurement values and the estimated values can be represented as:

$$e_M = x_M - \hat{x}_M \tag{11}$$

Finally, the detection indicator h_M is calculated as:

$$h_M = \sqrt{(e_M - \hat{\boldsymbol{\mu}})MCD^{-1}(e_M - \hat{\boldsymbol{\mu}})^T} \tag{12}$$

Using h_M as the test statistic, a fault is detected if $h_M > d$. The detection result can be used as decision support to adjust the turbine operation and maintenance accordingly.

C. PERFORMANCE METRICS

For performance evaluation and comparison, the well-established Area Under the Receiver Operating Characteristic (ROC) Curve (i.e., AUC) [34] is used in this study. The AUC is a comprehensive metric to evaluate the performance of fault detectors considering fault detection rate (FDR) and false alarm rate (FAR) under different confidence levels. FDR is defined as the percentage of fault samples detected over the fault period and FAR is the percentage of falsely identified fault samples over the normal operation period [28]. A good fault detection method will yield a high FDR while a low FAR. The AUC metric can provide a relative tradeoff between FDR and FAR. Generally, an AUC value close to one indicates an excellent performance of the detector, whereas an AUC equal to 0.5 indicates that the detector has no discriminative ability.

IV. CASE STUDY I: BENCHMARK SIMULATIONS

In this section, we evaluate our proposed MLD-AE algorithms for fault detection on a generic WT benchmark model.

A. BENCHMARK DESCRIPTION

Figure 3 shows the block diagram of the WT benchmark model. It models an offshore 5-MW three-bladed horizontal axis WT built upon fatigue, aerodynamics, structures, and turbulence (FAST) aeroelastic simulator. The model is driven

TABLE 1. Available sensor measurements in advanced benchmark.

Sensor Variable	Symbol	Units	Noise Power
Wind speed at hub height	$v_{w,m}$	m/s	0.0071
Rotor speed	$\omega_{r,m}$	rad/s	10^{-4}
Generator speed	$\omega_{g,m}$	rad/s	2×10^{-4}
Generator torque	$\tau_{g,m}$	Nm	0.9
Generated electrical power	$P_{g,m}$	rad/s	10
Pitch angle of ith Blade	$\beta_{i,m}$	deg(°)	1.5×10^{-4}
Azimuth angle low speed	ϕ_m	rad	10^{-3}
Blade root moment of ith Blade	$M_{B,i,m}$	Nm	10^3
Tower acceleration in x direction	x_m	m/s^2	5×10^{-4}
Tower acceleration in y direction	y_m	m/s^2	5×10^{-4}
Yaw error	Ξ_m	deg	5×10^{-2}

by the stochastic wind input and controlled in closed-loop with PI controllers. More detailed descriptions of the WT benchmark model can be found in [35].

In the benchmark model, a total of $m = 15$ sensor measurements are collected as listed in Table 1, where each measurement is added with a band-limited Gaussian white noise [35]. In this paper, we consider typical 5 fault scenarios including both sensor faults and actuator faults, as listed in Table 2. For each fault scenario simulation, the system begins to operate in normal conditions for 30 s, and then the fault is introduced and last for 20 s and finally cleared. Each fault scenario corresponds to an individual simulation. There are total 5 testing data sets corresponding to 5 fault scenarios. Each set contains both 2400 normal samples and 1600 fault samples, and all sets are tested for online fault detection. In addition, we generate 8000 normal samples as the training data used in the offline training phase. Note that the reported detection performances below are obtained from the average of 20 independent experiments to reduce the effects of the randomness.

B. PARAMETERS SETUP

For all model training, we use the stochastic gradient descent with a momentum for network optimization. We choose 8 hidden units to perform compressed representation learning. The learning rate and the momentum are set to 0.1 and 0.5, respectively. In order to speed up the training process, we split the training set into mini-batches to update the network weights, and the mini-batch size is 1000. The number of epochs during the training process is chosen as 3000. For all training data, each variable is linearly scaled to the range [0,1]. Accordingly, the testing data is rescaled according to the maximum and the minimum value of the training data, thus ensuring both data sets in the similar range. For DAE and MLD-AE, the hyperparameters (e.g. noise level υ) will be investigated in the following subsections. For online testing, different confidence levels are set in the range [0, 1] with a step size of 0.01 to calculate the AUC values.

C. ANALYSIS OF PARAMETERS IN MLD-AE

First, we examine the effects of different combinations of the initial noise level υ_0 and the final noise level υ_T.

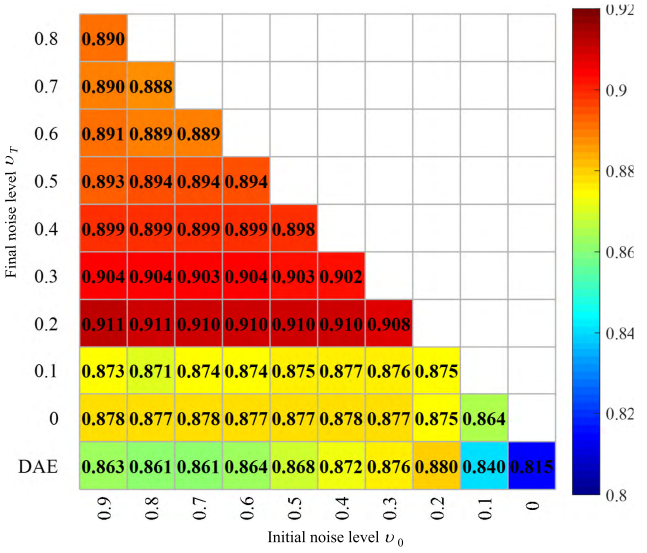

FIGURE 4. Parameters effects of initial noise level υ_0 and final noise level υ_T for MLD-AE-S1 with Zero-Masking noise evaluated on F5 testing dataset. The horizontal and vertical axis represent the initial noise level υ_0 and final noise level υ_T, respectively. Note that the bottom row corresponds to AUC values using DAE with different noise levels, and the initial noise level $\upsilon_0 = 0$ corresponds to the traditional AE.

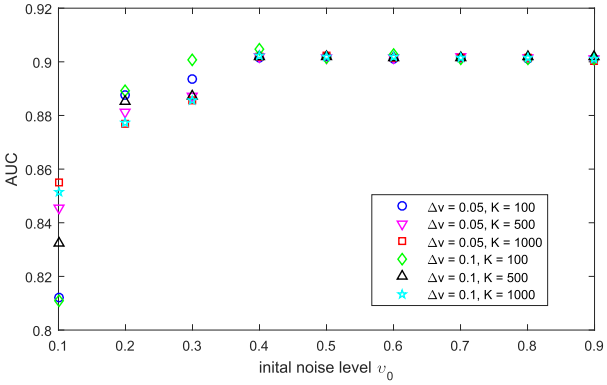

FIGURE 5. Parameters effects of $\Delta\upsilon$ and K for MLD-AE-S1 with Zero-Masking noise on detection performance evaluated on F5 testing dataset.

We considered $\upsilon_0 \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and the corresponding $\upsilon_T (\upsilon_T < \upsilon_0)$ can be chosen from the range [0,0.8]. The smallest possible final noise level is 0, which means no noise is added. The iteration steps $K = 500$ and the noise decreasing step size $\Delta\upsilon = 0.05$ in Algorithm 1 are used in this study. Figure 4 shows the evaluation results on F5 testing dataset (i.e., converter actuator fault) using MLD-AE-S1 with ZM noise. Clearly, for DAE, we can see that as the noise level υ increases, AUC value will first increase and then decrease from the noise level $\upsilon = 0.2$. This means that DAE achieves the best performance at the lower noise level $\upsilon = 0.2$. But the performance of DAE always outperforms that of traditional AE under all noise levels.

From Figure 4, we can see that the MLD-AE-S1 yields better performance in terms of AUC than the DAE. It is

TABLE 2. Fault scenarios considered in the advanced benchmark.

No.	Fault	Details	Type
F1	Generator Speed sensor	$w_{g,m}$ scaled by 1.1	Scaling
F2	Pitch angle sensor	Fixed Value on $\beta_{1,m}$ holds to 1 deg	Stuck
F3	Low speed shaft position encoder	random offset on ϕ_m	Bit error
F4	Blade Pitch actuator	$\omega_{n,2} = 5.73, \zeta_{n,2} = 0.45$	Abrupt Changes in Dynamics
F5	Torque converter	1000 Nm Offset on $\tau_{g,m}$	Offset

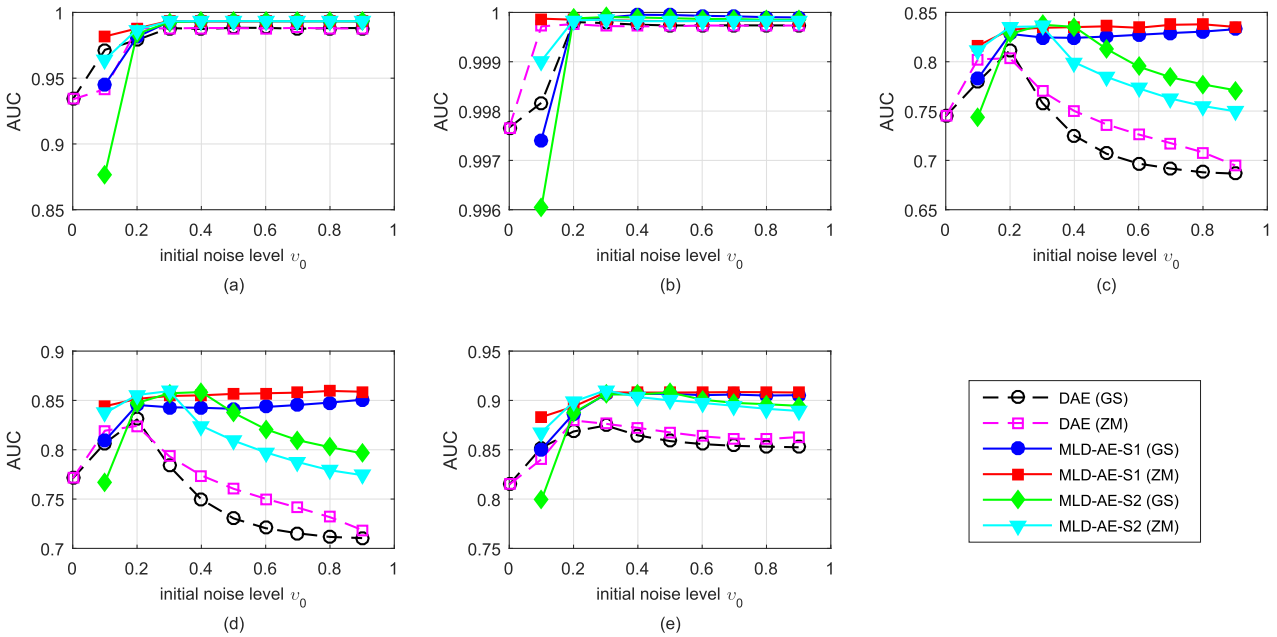

FIGURE 6. Comparison with different methods for testing datasets: (a) F1; (b) F2; (c) F3; (d) F4; (e) F5. Note that $\upsilon_0 = 0$ corresponds to traditional AE.

observed from each column in this figure that our proposed MLD-AE algorithm using multiple noise levels obtains larger AUC value than DAE only using a single fixed noise level. The largest AUC values is over 0.9, which demonstrates the effectiveness of our proposed method. Particularly, for MLD-AE-S1, it can be observed that when the final noise level υ_T is greater than the optimal noise level achieved by DAE, the performance will degrade. Besides, for all choices of the initial noise level υ_0, MLD-AE-S1 exhibits stable AUC values as long as the final noise level is not below the optimal noise level for DAE, which suggests MLD-AE-S1 is robust to the initial noise level υ_0.

Second, we investigate the effects of the parameters $(\Delta\upsilon, K)$, and the corresponding results evaluated on the F5 testing dataset are shown in Figure 5. Clearly, for all choices of $(\Delta\upsilon, K)$, MLD-AE-S1 obtains the relative consistent AUC value, when the initial noise level υ_0 is greater than 0.3. The result indicates our proposed MLD-AE-S1 method is robust to the parameters $(\upsilon_0, \Delta\upsilon, K)$ at higher initial noise levels. Note that for MLD-AE-S2, similar results have also been obtained.

D. PERFORMANCE COMPARISON

We evaluate the performance of our proposed MLD-AE-S1 and MLD-AE-S2 algorithms with ZM noise and GS noise, as well as the traditional DAE on all fault scenarios as listed in Table 2, and the corresponding results are shown in Figure 6, where AUC values for each method are calculated under different initial noise levels. Correspondingly, the best performance results for DAE and four MLD-AE methods are summarized in Table 3, where optimal noise parameters are presented in parentheses. The results obtained with the PCA method are also provided for comparison. For each fault scenario, the best performance among the considered methods has been highlighted in bold.

From Figure 6 and Table 3, it is clearly found that all four MLD-AE algorithms always perform better than DAE regardless of noise types and noise adding schemes, with the increase of the initial noise level υ_0. The significant improvement of detection performance can be observed in the cases of F3, F4 and F5, which are more difficult to detect compared to the scaling sensor fault (F1) and stuck sensor fault (F2). In the cases of F1 and F2, the stable performances are achieved when the initial noise level $\upsilon_0 > 0.2$. Meanwhile, it can be found that MLD-AE-S1 can obtain more stable and consistent performance at higher noise levels for most of the cases. This reduces the requirements on the choice of the initial noise levels. We also observe that MLD-AE-S1 with the ZM noise outperforms that with the GS noise.

TABLE 3. Comparative results on benchmark model in terms of AUC values.

No.	PCA	AE	DAE(GS)	DAE(ZM)	MLD-AE-S1(GS)	MLD-AE-S1(ZM)	MLD-AE-S2(GS)	MLD-AE-S2(ZM)
F1	0.7237	0.9339	0.9931 (0.4)	0.9931 (0.5)	0.9936 (0.5 → 0.3)	0.9936 (0.9 → 0.2)	**0.9937** (0.3 → 0.2)	0.9934 (0.4 → 0.1)
F2	0.9815	0.9977	0.9999 (0.3)	0.9999 (0.3)	**1** (0.3 → 0.2)	**1** (0.3 → 0.2)	0.9999 (0.3 → 0.2)	0.9999 (0.3 → 0.2)
F3	0.5190	0.7449	0.8369 (0.3)	0.8284 (0.3)	0.8329 (0.3 → 0.2)	0.8323 (0.5 → 0.2)	**0.8387** (0.3 → 0.1)	0.8358 (0.2 → 0.1)
F4	0.5321	0.7664	0.8565 (0.3)	0.8496 (0.3)	0.8507 (0.5 → 0.2)	8596 (0.2 → 0.1)	**0.8632** (0.3 → 0.1)	0.8595 (0.3 → 0.2)
F5	0.5642	0.8147	0.9016 (0.3)	0.8992 (0.3)	0.9021 (0.4 → 0.3)	0.9096 (0.5 → 0.2)	0.9066 0.3 → 0.2)	**0.9102** (0.3 → 0.1)

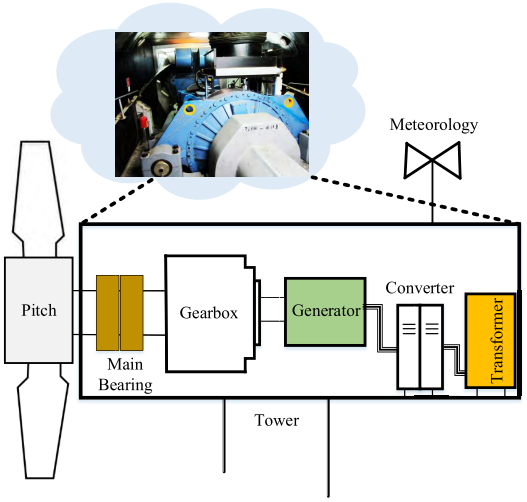

FIGURE 7. Schematic diagram of a real WT.

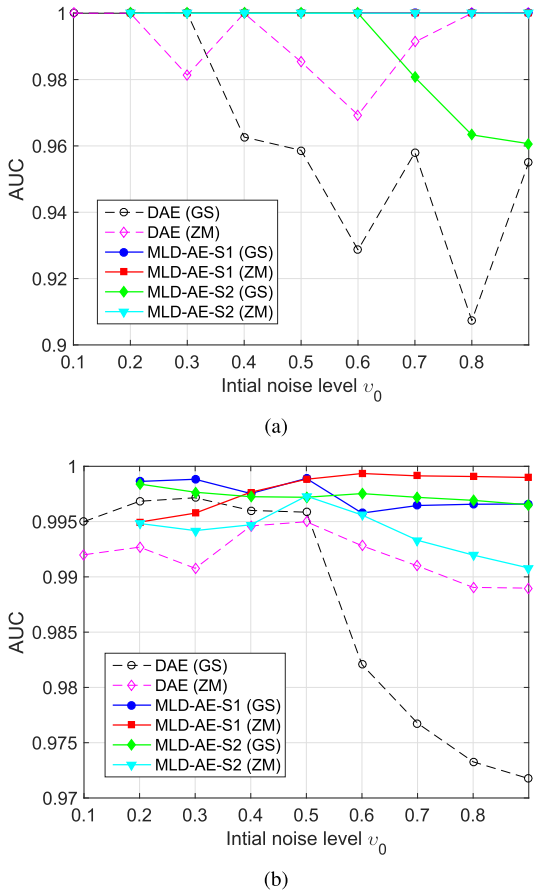

FIGURE 8. Performance comparison for two different fault scenarios (a) T1; (b) T2.

To check more details on fault detection performance of the proposed method, we calculate both metrics of FDR and FAR for all five fault cases with different detection methods, and the comparative results ares listed in Table 4, where the confidence level α for detection calculation is set to 0.95. For all cases, our proposed MLD-AE methods achieved larger FDR values and smaller FAR values than traditional methods of DAE, AE and PCA, especially for the cases of F3, F4, and F5. This further demonstrates the superiority of the proposed method.

V. CASE STUDY II: REAL SCADA DATA

In this section, the proposed method is further evaluated by using SCADA data collected from a real operational wind farm in Inner Mongolia in China. The wind farm includes over 100 variable-speed pitch-controlled WTs with a multi-stage gearbox and a doubly-fed induction generator. Figure 7 shows the structure sketch of a WT. All turbines are equipped with the standard SCADA system to monitor and control the operational status of turbines. The SCADA data are collected at 30-s intervals and mainly consists of over 100 readings, including temperatures, pressures, vibrations, power outputs, wind speed and digital controls. In this case study, similar to benchmark models described in IV, we select relevant 11 sensor variables for fault detection performance evaluation as listed in Table 5. According to the maintenance documents, two different fault scenarios occurred in different turbines are considered and the detailed information about these two faults are listed in Table 6.

A. DATA PREPROCESSING

Practically, there are usually some bad data in the historical operational SCADA data, including outlier points and invalid values, maybe due to acquisition errors, communication problems or other issues. Additionally, there exist some gaps with no power generation when a WT is inactive during periods of lower and higher wind speeds, and due to the occurrence of downtime and maintenance periods. Therefore, it is necessary to remove these bad data and gaps prior to model training. Eventually, only those data under generating power condition of WTs are retained.

TABLE 4. Comparative results on benchmark model in terms of FDR and FAR.

Methods	FDR					FAR				
	F1	F2	F3	F4	F5	F1	F2	F3	F4	F5
PCA	0.1281	0.9931	0.0931	0.1037	0.1056	0.0754	0.0754	0.0754	0.0754	0.0754
AE	0.7129	0.9994	0.1588	0.1781	0.3958	0.0683	0.0683	0.0682	0.0684	0.0684
DAE(GS)	0.9562	0.9994	0.1634	0.2006	0.5164	0.0472	0.0475	0.0473	0.0472	0.0471
DAE(ZM)	0.9568	0.9994	0.1725	0.2105	0.5084	0.0437	0.0438	0.0438	0.0437	0.0437
MLD-AE-S1(GS)	0.9850	0.9994	0.2383	0.2733	0.5859	0.0455	0.0458	0.0459	0.0455	0.0458
MLD-AE-S1(ZM)	0.9788	0.9994	0.4434	0.4900	0.6407	0.0388	0.0390	0.0388	0.0392	0.0389
MLD-AE-S2(GS)	0.9772	0.9994	0.5070	0.5871	0.6226	0.0257	0.0252	0.0252	0.0251	0.0251
MLD-AE-S2(ZM)	0.9838	0.9994	0.2810	0.3195	0.5806	0.0409	0.0406	0.0409	0.0411	0.0412

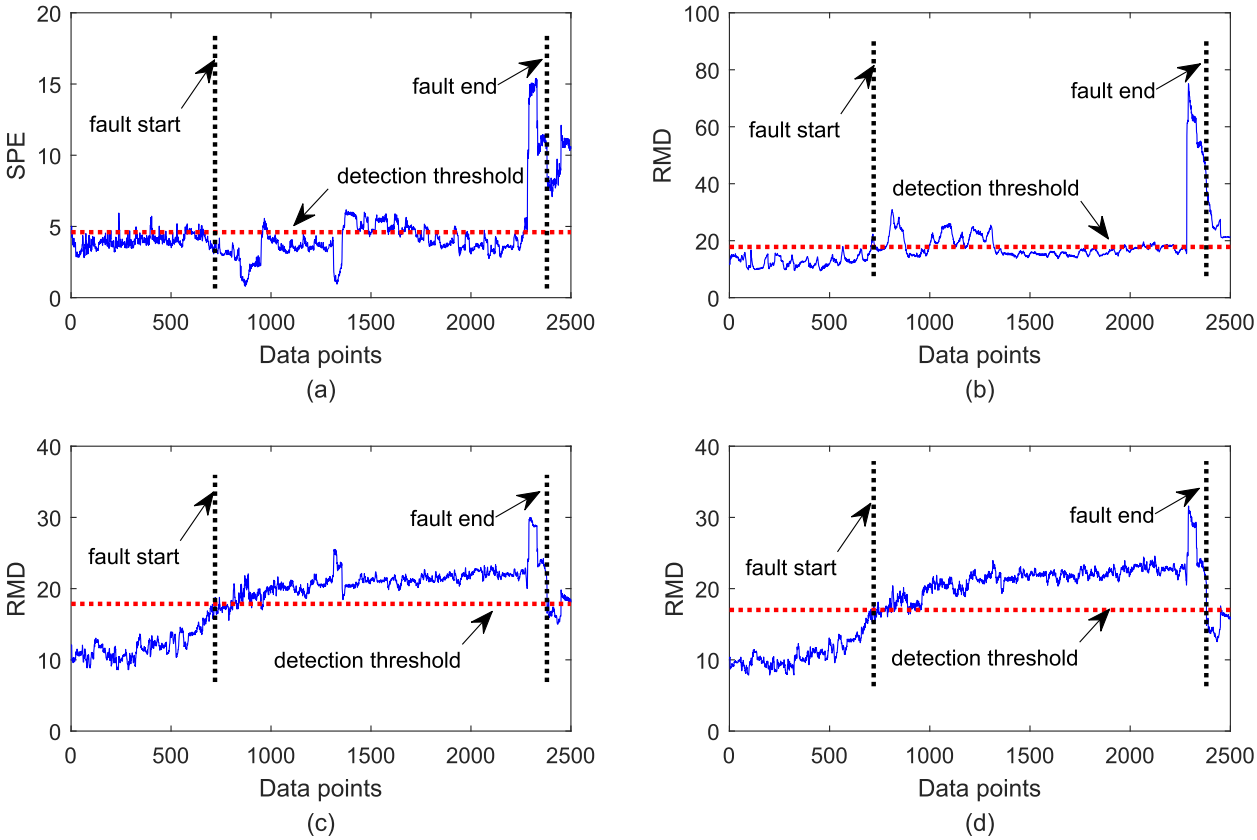

FIGURE 9. Performance comparison on drive train vibation anomaly detection using (a) PCA; (b) AE; (c) DAE(ZM); (d) MLD-AE-S1 (ZM).

TABLE 5. SCADA variables used in this study.

Sensor Variable	Unit	Sensor Variable	Unit
Wind Speed	m/s	Generated Power	kW
Rotor Speed	r/min	Gearbox Speed	r/min
Generator Speed	r/min	Generator Torque	Nm
Blade 1 Pitch angle	deg(°)	Tower acceleration	m/s^2
Blade 2 Pitch angle	deg(°)	Drive train acceleration	m/s^2
Blade 3 Pitch angle	deg(°)		

Then, the training and testing datasets are chosen from the preprocessed SCADA data, where the training set contains 8000 normal samples and the testing set contains 2000 normal samples and 2000 fault samples. All data are also linearly scaled to the range to [0,1] before model training to ensure that all sensor variables lies in a similar range.

TABLE 6. Fault Scenarios considered on SCADA Data.

No.	Fault Description	Fault start time	Fault end time
T_1	Pitch System Fault	2014/09/19 16:47:24	2014/09/20 02:50:31
T_2	Drive train vibration anomaly	2014/05/22 14:39:24	2014/05/22 23:36:13

B. RESULTS ANALYSIS

The performance evaluation results for two considered fault scenarios using different methods are presented in Figure 8. Table 7 summarizes the best performance of DAE and MLD-AE methods on optimal noise level parameters. Clearly, our proposed MLD-AE algorithms generally outperform the traditional DAE, AE and linear PCA methods.

TABLE 7. Comparative results on SCADA Data in terms of AUC.

No.	PCA	AE	DAE (GS)	DAE (ZM)	MLD-AE-S1 (GS)	MLD-AE-S1 (ZM)	MLD-AE-S2 (GS)	MLD-AE-S2 (ZM)
T_1	0.8940	0.9658	**1** (0.1)	**1** (0.1)	**1** (0.5)	**1** ($0.2 \rightarrow 0.1$)	**1** ($0.2 \rightarrow 0.1$)	**1** ($0.2 \rightarrow 0.1$)
T_2	0.9053	0.9627	0.9972 (0.4)	0.9966 (0.1)	0.9989 ($0.5 \rightarrow 0.1$)	**0.9993** ($0.6 \rightarrow 0.1$)	0.9984 ($0.2 \rightarrow 0.1$)	0.9973 ($0.5 \rightarrow 0.1$)

Specifically, DAE with the ZM noise exhibits better performance than DAE with the GS noise, and the latter shows the significant performance degradation and bigger variations at higher noise levels. The main reason is that adding high level Gaussian noise will mask useful information contained in original data, thus affecting detection results. In contrast, the MLD-AE method presents relatively consistent detection performance for two fault scenarios.

Furthermore, it can be seen from Table 7 that the first fault scenario is more easily detected with the AUC value of 1 for all DAE-based and MLD-AE methods. This means the pitch system fault can be successfully detected with no false alarms. For the second fault case, all DAE-based and MLD-AE methods obtain the large FAR values of over 0.99 and significantly outperformed traditional AE and PCA methods. In order to further check that whether the proposed method can detect the fault start point and the fault end point, we calculate the fault indicators of the testing data covering the fault period of drive train vibration anomaly listed in Table 6, and the comparative results obtained using four representative methods are given in Figure 9. In this figure, the detection threshold is determined with the confidence level $\alpha = 0.95$. From Figure 9, the DAE-based and MLD-AE-based methods accurately identify the fault start point and the fault end point, and during the fault period, the fault indicators exceed the detection thresholds, suggesting a large FDR value. Compared with MLD-AE, the AE produces a small number of missed detections. Notably, both AE and PCA methods performed worse with a large number of missed detections during the fault period, and even worse, they cannot detect the occurrence of the fault in time. This result further demonstrate the superiority of the proposed method.

VI. CONCLUSIONS

This paper presented a general multivariate data-driven fault detection framework for WTs based on nonlinear autoencoder neural networks. The proposed framework offline builds a normal behavior model using multivariate normal data, and then online identifies potential faults by comparing fault indicators derived from residuals. In this framework, a new MLD-AE method was proposed to discover more useful patterns and capture nonlinear dependencies from multivariate noisy sensor data. Compared with traditional approaches, such as DAE, AE and PCA, our proposed method achieved better fault detection performance in terms of AUC, FDR and FAR on both simulated and real case studies. In addition, our proposed method can more accurately and timely the fault occurrence with a larger FDR and a smaller FAR, which has great potentials for practical applications.

ACKNOWLEDGMENT

X. Li was with the School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China.

REFERENCES

[1] P. Tavner, *Offshore wind Turbines: Reliability, Availability and Maintenance* (Energy Engineering). Edison, NJ, USA: IET, 2012.
[2] W. Qiao and D. Lu, "A survey on wind turbine condition monitoring and fault diagnosis—Part I: Components and subsystems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6536–6545, Oct. 2015.
[3] X. Wei, M. Verhaegen, and T. van Engelen, "Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques," *Int. J. Adapt. Control Signal Process.*, vol. 24, no. 8, pp. 687–707, Aug. 2010.
[4] L. Zhang and Z.-Q. Lang, "Wavelet energy transmissibility function and its application to wind turbine bearing condition monitoring," *IEEE Trans. Sustain. Energy*, vol. 9, no. 4, pp. 1833–1843, Oct. 2018.
[5] J. Li, M. Li, X. Yao, and H. Wang, "An adaptive randomized orthogonal matching pursuit algorithm with sliding window for rolling bearing fault diagnosis," *IEEE Access*, vol. 6, pp. 41107–41117, 2018.
[6] G. Jiang, H. He, P. Xie, and Y. Tang, "Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2391–2402, Sep. 2017.
[7] Y. Qin, J. Zou, and F. Cao, "Adaptively detecting the transient feature of faulty wind turbine planetary gearboxes by the improved kurtosis and iterative thresholding algorithm," *IEEE Access*, vol. 6, pp. 14602–14612, 2018.
[8] D. Wu, W. Liu, J. Song, and Y. Shen, "Fault estimation and fault-tolerant control of wind turbines using the SDW-LSI algorithm," *IEEE Access*, vol. 4, pp. 7223–7231, 2016.
[9] G. Q. Jiang, P. Xie, X. Wang, M. Chen, and Q. He, "Intelligent fault diagnosis of rotary machinery based on unsupervised multiscale representation learning," *Chin. J. Mech. Eng.*, vol. 30, no. 6, pp. 1314–1324, Nov. 2017.
[10] B. Yang, R. Liu, and X. Chen, "Sparse time-frequency representation for incipient fault diagnosis of wind turbine drive train," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 11, pp. 2616–2627, Nov. 2018.
[11] W. Yang, R. Court, and J. Jiang, "Wind turbine condition monitoring by the approach of SCADA data analysis," *Renew. Energy*, vol. 53, pp. 365–376, May 2013.
[12] A. Kusiak and W. Li, "The prediction and diagnosis of wind turbine faults," *Renew. Energy*, vol. 36, no. 1, pp. 16–23, Jan. 2011.
[13] P. Bangalore and L. B. Tjernberg, "An artificial neural network approach for early fault detection of gearbox bearings," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 980–987, Mar. 2015.
[14] Y. Zhang, C. Zhang, Y. Zhao, and S. Gao, "Wind speed prediction with RBF neural network based on PCA and ICA," *J. Electr. Eng.*, vol. 69, no. 2, pp. 148–155, Mar. 2018.
[15] Y. Zhang, B. Chen, Y. Zhao, and G. Pan, "Wind speed prediction of IPSO-BP neural network based on lorenz disturbance," *IEEE Access*, vol. 6, pp. 53168–53179, 2018.
[16] C. Yang, J. Liu, Y. Zeng, and G. Xie, "Real-time condition monitoring and fault detection of components based on machine-learning reconstruction model," *Renew. Energy*, vol. 133, pp. 433–441, Apr. 2019.
[17] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018.
[18] 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.
[19] 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.
[20] M. Kruger, S. X. Ding, A. Haghani, P. Engel, and T. Jeinsch, "A data-driven approach for sensor fault diagnosis in gearbox of wind energy conversion system," in *Proc. IEEE 10th Int. Conf. Control Autom. (ICCA)*, Jun. 2013, pp. 227–232.

[21] F. Pozo and Y. Vidal, "Wind turbine fault detection through principal component analysis and statistical hypothesis testing," *Energies*, vol. 9, no. 1, p. 3, Jan. 2016.
[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[23] W. Qiao, R. G. Harley, and G. K. Venayagamoorthy, "Fault-tolerant indirect adaptive neurocontrol for a static synchronous series compensator in a power network with missing sensor measurements," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1179–1195, Jul. 2008.
[24] Z. Meng, X. Zhan, J. Li, and Z. Pan, "An enhancement denoising autoencoder for rolling bearing fault diagnosis," *Measurement*, vol. 130, pp. 448–454, Dec. 2018.
[25] Z. S. Vanini, N. Meskin, and K. Khorasani, "Multiple-model sensor and components fault diagnosis in gas turbine engines using autoassociative neural networks," *J. Eng. Gas Turbines Power*, vol. 136, no. 9, May 2014, Art. no. 091603.
[26] L. Wang, Z. Zhang, J. Xu, and R. Liu, "Wind turbine blade breakage monitoring with deep autoencoders," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2824–2833, Jul. 2018.
[27] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 1096–1103.
[28] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 89–100, Feb. 2018.
[29] 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, Dec. 2010.
[30] K. J. Geras and C. Sutton. (2014). "Scheduled denoising autoencoders." [Online]. Available: https://arxiv.org/abs/1406.3269
[31] Y. Wang, Q. Miao, E. W. Ma, K.-L. Tsui, and M. G. Pecht, "Online anomaly detection for hard disk drives based on Mahalanobis distance," *IEEE Trans. Rel.*, vol. 62, no. 1, pp. 136–145, Mar. 2013.
[32] J. Hardin and D. M. Rocke, "The distribution of robust distances," *J. Comput. Graph. Statist.*, vol. 14, no. 4, pp. 928–946, Jan. 2012.
[33] P.-E. P. Odiowei and Y. Cao, "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 36–45, Feb. 2010.
[34] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2008.
[35] P. F. Odgaard and K. E. Johnson, "Wind turbine fault detection and fault tolerant control-An enhanced benchmark challenge," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 4447–4452.

XIN WU received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2000, and the M.S. degree from Beihang University, Beijing, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Electrical Engineering, Yanshan University, Qinhuangdao, China. His research interests include big data analytics, data mining, and prognostics and health management of wind energy systems.

GUOQIAN JIANG (M'18) received the B.S. degree in measurement control technology and instrumentation and the Ph.D. degree in control science and engineering from Yanshan University, Qinhuangdao, China, in 2011 and 2018, respectively. He was a joint Ph.D. student with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA, from 2015 to 2017.

He is currently a Lecture with the School of Electrical Engineering, Yanshan University. His research interests include advanced signal processing algorithms, intelligent fault diagnostics and prognostics, and deep learning for machine health monitoring.

XIAO WANG received the B.S. degree in measurement control technology and instrumentation and the M.S. degree in detection technology and automatic equipment from Yanshan University, Qinhuangdao, China, in 2003 and 2005, respectively, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering. His research interests include signal processing, fault diagnosis, and port automation.

PING XIE received the M.S. degree in measuring technology and instrumentation and the Ph.D. degree in circuits and systems from Yanshan University, Qinhuangdao, China, in 1996 and 2006, respectively.

Since 2009, she has been a Professor with the School of Electrical Engineering, Yanshan University. Her research interests include multi-modal information processing and fusion, brain-computer interface, rehabilitation robot biofeedback control, wearable equipment development, and condition monitoring and fault diagnosis of complex systems.

XIAOLI LI received the B.S.E. and M.S.E. degrees from the Kunming University of Science and Technology, and the Ph.D. degree from the Harbin Institute of Technology, China, in 1992, 1995, and 1997, respectively, all in mechanical engineering.

From 1998 to 2009, he was with the City University of Hong Kong as a Research Fellow; an Alexander von Humboldt Scholar of Hannover University, Germany; and a Postdoctoral Fellow of The Chinese University of Hong Kong and the School of Computer Science, University of Birmingham, U.K. From 2009 to 2011, he was appointed as a Professor and the Head of the Department of Automation, Institute of Electrical Engineering, Yanshan University, China. Since 2011, he has been a Full Professor and the Vice Director of the National Key Laboratory of Cognitive Neuroscience and Learning and the Director of the Center for Neuromodulation and Cognitive Enhancement, Beijing Normal University, China. He is a co-founder of the International Consortium of Autism Institutes. His main areas of research interests include neural engineering, computation intelligence, signal processing and data analysis, monitoring system, and manufacturing system. He is the author of 65 Chinese patents, 256 journal papers (32 Chinese papers), 26 conference papers, one English book (*Signal Processing in Neuroscience* 2016), and two Chinese books. His post and present research projects include the National Science Fund for Distinguished Young Scholars and the National Natural Science Foundation of China.

Dr. Li is an Editor of the *Frontiers in Neuroscience*, the *Frontiers in Computational Neuroscience*, the *BMC Neurology*, and the *Transnational Neuroscience*.