Intelligent Fault Classification Exploration Inspired by Suprathreshold Stochastic Resonance

Ziheng Xu, Siliang Lu, *Senior Member, IEEE*, Yanmei Kang, and Jun Jiang

***Abstract*—The aim of this article is to present a novel intelligent fault diagnosis method based on the principle of stochastic resonance (SR) in array of noisy ReLU activators. To this end, the phenomenon of suprathreshold stochastic resonance (SSR), which is a resonance like collective effect of the array model for the input above threshold, is demonstrated, and the limit of the input-output relation of the average response is explicitly acquired. With the explicit noise intensity dependent relation as an activation function, the new fault diagnosis algorithm is proposed by training the intensity of Gaussian white noise as an independent parameter, and the corresponding initial value is set referring to the minimum point of the loss function defined by cross-entropy. A penalty term is further introduced to the loss function to ensure the nonnegativity of noise intensity. The performance and noise robustness of the proposed algorithms is validated by typical state-of-the-art networks across three open datasets, namely, CWRU bearing fault dataset, SEU gearbox dataset, and HUST varying speed gearbox dataset. It is found that the proposed algorithm outperforms its counterparts both in the absence of noise and in the presence of fixed noise on test accuracy. Particularly, the proposed algorithm can markedly improve the test accuracy when the training samples are limited and noise contaminated.**

***Index Terms*—Intelligent fault diagnosis, limit of the input-output relation, noise exploitation, noisy ReLU array, suprathreshold stochastic resonance (SSR).**

I. Introduction

With the high development of modern industry and automatic technology, rotating machinery devices, including wind turbines [1], aircraft engines [2], and industrial robots [3], tend to run under more and more sophisticated working conditions and suffer from increasing environmental erosion. To ensure the reliability and safety of productivity [4], it is indispensable to conduct real-time fault monitoring, so that appropriate actions for prevention and maintenance can be taken timely. Driven by this substantial practical demand, how to develop effective methods for diagnosing rotating machinery faults has remained a sustained and active field during the recent decades.

Received 26 May 2025; revised 13 July 2025; accepted 23 July 2025. Date of publication 11 August 2025; date of current version 21 August 2025. This work was supported by the National Nature Science Foundation of China under Grant 12172268. The Associate Editor coordinating the review process was Dr. Hongtian Chen. *(Corresponding author: Yanmei Kang.)*

Ziheng Xu and Yanmei Kang are with the Department of Applied Mathematics, School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: ymkang@xjtu.edu.cn).

Siliang Lu is with the College of Electrical Engineering and Automation, Anhui University, Hefei 230601, China.

Jun Jiang is with the State Key Laboratory for Strength and Vibration, Xi'an Jiaotong University, Xi'an 710049, China.

Digital Object Identifier 10.1109/TIM.2025.3597686

There have been various fault diagnosis methods, among which two categories can be classified. One category is to boost the signal-to-noise ratio (SNR) of the fault component via classical signal processing techniques, such as the Hilbert–Huang transform [5], [6], wavelet analysis [7], [8], and multiple bandpass filter [9], or anti-intuitive signal processing methods based on nonlinear phenomena, such as stochastic resonance (SR) [10], [11], [12], [13], [14] and bifurcation dynamics [15]. Particularly, as a counterintuitively cooperative effect between noise and weak signal in certain nonlinear circumstance [16], the principle of SR has been proven effective in detecting incipient mechanical faults. Nevertheless, the classical noise reduction-based methods tend to reduce the useful information of the involving faults, while the anti-intuitive frequency enhancement-based methods might fail to identify its severity. The other category is the so-called intelligent diagnosis [17], [18], [19], [20], which performs fault monitoring using machine learning. Since the intelligent fault diagnosis methods can simultaneously recognize the temporal and spatial pattern through self-learning, it should be able to outperform the first category methods in detecting feeble and complex fault signal even under sophisticated working conditions.

Currently, as the deep-learning technology thrives in the era of AI, various intelligent fault diagnosis techniques, including adaptive deep belief networks [21], 1-D convolution neural network (1D-CNN) [22], two-dimensional gate recurrent unit (GRU)-based network [23], dual-path convolution network with attention mechanism and bidirectional gated recurrent unit (DCA-BIGRU) [24], and deep residual shrinkage networks (DRSNs) [25], have been proposed for better diagnosis performance. Nevertheless, more complex backbone usually means more computing power demand. Thus, differing from continuously complicating the structure of neural networks as usual, in this article, we aim to keep the existing network backbones unchanged but update activation function via noise utilization and array enhancement.

Adding noise into the design of machine learning can be dated back to Bishop's seminar research [26]. The injected noise can be utilized toward improving the generalization ability [27], [28], softening hard threshold [29], [30], boosting the training accuracy [31] and robustness [32], enhancing unsupervised representation [33], and speeding up the training process [34]. More recently, similar noise benefit has been adopted in more diverse fields, such as generative model [35], epilepsy diagnosis [36], pretrained vision-language models [37], and federal learning [38]. Nevertheless, to the best of

our knowledge, in the field of intelligent fault diagnosis, noise is conventionally treated as an annoying factor [5], [6], [7], [8], [39]. Although the injected noise was found beneficial in a few recent researches [40], [41], [42]. Note that in [40], [41], and [42], the involving noise intensity was still treated as hyperparameter or tuned based on the experience from the pre-trained phase, which is not convenient for practical use. This insufficiency motivates us to develop new intelligent diagnosis algorithm by using noise intensity as trainable parameter, so that its optimal value can be automatically achieved.

Actually, the training of noise intensity can also be validated by the principle of SR or array enhanced SR [30], [43], [44], [45]. Note that weak signals can generally be distinguished as subthreshold and suprathreshold. The former is typical for classical SR, while the latter is associated with suprathreshold stochastic resonance (SSR) [46], [47], [48]. The classical SR can occur in single threshold activator, but the SSR can only emerge in the array of threshold activators. In fact, the SSR is more important for machinery fault diagnosis, since signals from real-world applications are always the mixture of suprathreshold and subthreshold components. Inspired by the phenomenon of SSR, we adopt the array of noisy ReLU activators as basic processing unit in the method of this article. Here, the reason that the ReLU activator is adopted rather than the others lies in that the ReLU activator is typical in deep learning [49], [50], [51], and we are confident that similar findings can be discovered if the ReLU activator is replaced by another activator.

Note that the role of activator or activation function in machine learning is equal to the role of sensor or detector in signal processing; hence, these terminologies can be equivalently employed from now on. It is not hard to imagine that the machine time occupied by neural network with the array of noisy ReLU sensors as nodes can exceed far beyond the machine time occupied by the network of the same architecture but with single noisy ReLU activator [49] as nodes, though. To evade this disadvantage, we adopt the limit of the input-output relation of the large-sized array of ReLU sensors as activation function. This operation is remarkably different from the existing fault diagnosis literature [40], [41], and it can also be explained by law of large number.

The novelty of this artcile can be summarized as follows. First, the phenomenon of SSR in the array of noisy ReLU activators is illustrated. Second, the input-output relation is explicitly deduced and adopted as activation function of our intelligent diagnosis method, so that the noise intensity can be trained independently toward the maximal utilization of noise benefit. With this technical novelty in mind, the new method is validated on distinct network architectures across different public datasets. It is found dramatically superior to the existing methods of the same network architecture with deterministic or noisy ReLU activation, especially when the datasets are noise contaminated or the training samples are of small size.

This article is organized as follows. The phenomenon of SSR is exhibited in the array model of noisy ReLU activators as principal demonstration in Section II. In Section III, the novel fault diagnosis algorithms are proposed by taking the limit of the average input-output relation of the ReLU array as activation function, and the learning rules via the back-propagation learning are also given. In Section IV, the performance of our algorithm is validated on typical network architectures with several datasets. In Section V, the noise robustness of the proposed algorithm is examined. Finally, conclusions are drawn in Section VI.

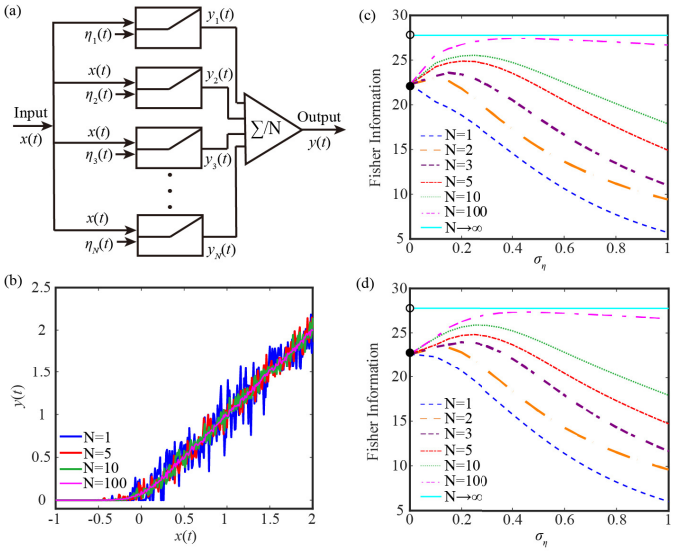


Fig. 1. (a) Scheme diagram of the ReLU array (1) and its SSR effect: the scheme diagram. (b) Average input-output relation. (c) Fisher information with the inner race faulty signal. Fisher information with the outer race faulty signal. In (c) and (d), the empty circle marks the fisher information achieved at $N \to \infty$ for $\sigma_\eta \neq 0$, while the solid circle shows the fisher information at $\sigma_\eta = 0$ for any finite N.

II. Suprathreshold Stochastic Resonance

Let us consider a parallel array of N uncoupled identical ReLU sensors, as shown in Fig. 1(a), each sensor having the same input-output relation

$$Y_i(t) = g(X_i(t)), \quad X_i(t) = X(t) + \eta_i(t) \tag{1}$$

for $i = 1, \ldots, N$. Here $X(t) = s_a(t) + \xi(t)$ is the input mixture of a parameterized signal $s_a(t)$ of parameter a and a noise term $\xi(t)$, which is the additive Gaussian white noise of standard deviation σ_ξ, and $\eta_i(t)$ is the Gaussian white noise of standard deviation σ_η, representing the measurement noise in the ith sensor. In the parallel array, $\eta_i(t), 1 \leq i \leq N$ are usually assumed mutually independent, i.e., $\langle \eta_i(t)\eta_j(s)\rangle = \sigma_\eta^2 \delta_{ij}\delta(t-s)$, where $\delta(\cdot)$ is the Dirac delta function and δ_{ij} being the Dirac notion. That is, all the measurement noise terms are independent identical distributed. In (1), $g(x)$ is a nonlinear function that transforms the continuous input into a discrete or semi-discrete output and we take $g(x)$ as the ReLU function to keep up with the subsequent algorithms in this article, namely, $g(x) = xH(x)$ with $H(\cdot)$ being Heaviside function. It is worthy to remark that we happen to select this typical activation function in this article and it can totally be replaced by any other activator.

Note that in (1), $Y_i(t)$ is the output of the ith sensor, and the mean response of this array is usually denoted as $\bar{Y}(t) = (1/N)\sum_{i=1}^{N} Y_i(t)$. In fact, we can interpret the array

(1) following technical terms from statistics. Let $Y(t) = g(X(t) + \eta(t))$ be an ensemble, then $(Y_1(t), Y_2(t), \ldots, Y_n(t))$ stands for a simple random sample from the ensemble and $\bar{Y}(t)$ is its sample mean. By law of large number, $\bar{Y}(t)$ converges in probability to $E[Y(t)]$, the ensemble mean. To enhance the readers' readability in this section, let us use capital letters for random variables and lowercase letters for their values.

To demonstrate the phenomenon of SR, let us use the Fisher information [47], [52] as the quantifying index. Let $p_{\bar{Y}}(y)$ be the probability density of the array output $\bar{Y}(t)$, then the Fisher information [53] can be defined as

$$J_{\bar{Y}} = \int_{-\infty}^{+\infty} \frac{1}{p_{\bar{Y}}(y)} \left(\frac{\partial}{\partial a} p_{\bar{Y}}(y) \right)^2 dy. \tag{2}$$

Let $p_X(x)$ be the probability density for the mixed input $X(t) = s_a(t) + \xi(t)$. Note that $\xi(t) \sim N(0, \sigma_\xi)$, then

$$p_X(x) = p_\xi(x - s_a(t)) = \frac{1}{\sqrt{2\pi}\sigma_\xi} e^{-\frac{(x - s_a(t))^2}{2\sigma_\xi^2}}. \tag{3}$$

Then, by the whole probability formula for density function

$$p_{\bar{Y}}(y) = \int_{-\infty}^{+\infty} p_{\bar{Y}|X}(y|x)\, p_X(x)\, dx \tag{4}$$

and

$$\frac{\partial}{\partial a} p_{\bar{Y}}(y) = -\frac{\partial s_a}{\partial a} \int_{-\infty}^{+\infty} p_{\bar{Y}|X}(y|x)\, p'_\xi(x - s_a)\, dx. \tag{5}$$

Thus, this probability density $p_{\bar{Y}}(y)$ and Fisher information $J_{\bar{Y}}$ can be obtained if the conditional probability density $p_{\bar{Y}|X}(y|x)$ is acquired. To do so, we introduce the notion $\bar{Y}|_{X=x}$ to denote the conditional random variable defined by $\bar{Y}$ under additional condition $X(t) = x$. Then, we will explain how to acquire $p_{\bar{Y}|X}(y|x)$ in two cases.

First, we consider the case where N is finite but not so large for central limit theorem to be applied. For the simplicity in notation, let us denote $Z_i = Y_i|_{X=x} \geq 0$ for $1 \leq i \leq N$. Note that for $z \in R$

$$\begin{aligned} P(Z_i \leq z) &= P(x + \eta_i \leq z)\, I_{\{z>0\}}(z) \\ &\quad + P(x + \eta_i \leq 0)\, I_{\{z=0\}}(z) \\ &= F_\eta(z - x)\, I_{\{z>0\}}(z) + F_\eta(-x)\, I_{\{z=0\}}(z) \end{aligned} \tag{6}$$

where $I_A(x)$ is the indicator function of the set A and $F_\eta(x) = (1/(2\pi)^{1/2}\sigma_\eta) \int_{-\infty}^{x} e^{-(u^2/2\sigma_\eta^2)} du$ is the accumulative distribution function of $\eta(t)$. Differentiating the both sides of (6) yields

$$p_{Z_i}(z) = p_\eta(z - x)\, I_{\{z>0\}} + F_\eta(-x)\, \delta(z) \tag{7}$$

with $p_\eta(x) = F_\eta{}'(x)$ being the probability density. Note the characteristic function of one random variable is defined by the Fourier transform of its probability density. The characteristic function of $\eta(t)$ reads

$$\tilde{p}_\eta(\omega) = \int_{-\infty}^{+\infty} e^{j\omega x} p_\eta(x)\, dx = \exp\left(-\frac{1}{2}\sigma_\eta^2 \omega^2 \right) \tag{8}$$

with $j^2 = -1$. Let $\tilde{p}_{Z_i}(\omega) = F\{p_{Z_i}(z)\}(\omega)$ be the Fourier transform of $p_{Z_i}(z)$, then

$$\begin{aligned} \tilde{p}_{Z_i}(\omega) &= \int_0^\infty e^{j\omega z} p_\eta(z - x) dz + F_\eta(-x) \\ &= e^{j\omega x} \int_{-x}^{\infty} e^{j\omega u} p_\eta(u) du + 1 - \Phi\left(\frac{x}{\sigma_\eta} \right) \end{aligned} \tag{9}$$

where Φ is the cumulative distribution function of standard normal variable. Note that the probability density of the sum of independent random variables equals to the multidimensional convolution of individual probability densities. Since Z_i is independent identical distributed for $1 \leq i \leq N$, the characteristic function of the sum $S = \sum_{i=1}^N Z_i$ can be acquired as $\prod_{i=1}^N \tilde{p}_{Z_i}(\omega)$. Let $f_{\bar{Z}}(z)$ and $f_S(s)$ be the probability density for the sample mean $\bar{Z} = (1/N)\sum_{i=1}^N Z_i$ and the sum S, respectively, then $f_{\bar{Z}}(z) = N f_S(Nz)$. Then, by the scaling property of the Fourier transform, it is not hard to obtain the characteristic function for $\bar{Z}$ as $\prod_{i=1}^N \tilde{p}_{Z_i}((\omega/N))$. Thus, the conditional probability density of $\bar{Y}|_{X=x} = \bar{Z}$ can be calculated by the inverse Fourier transform

$$\begin{aligned} p_{\bar{Y}|X}(y \mid x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Bigg(& e^{j\omega x/N} \int_{-x}^{\infty} e^{j\omega u/N} p_\eta(u)\, du \\ & + 1 - \Phi\left(\tfrac{x}{\sigma_\eta} \right) \Bigg)^N e^{-j\omega y}\, d\omega. \end{aligned} \tag{10}$$

It is clear that (9) and (10) can be numerically obtained by the Fourier transform and the inverse Fourier transform.

Second, we consider the case of sufficiently large N. By the central limit theorem, $\bar{Y}|_{X=x}$ approximately obeys normal distribution. That is, the following approximation holds:

$$p_{\bar{Y}|X}(y \mid x) = \frac{1}{\sqrt{2\pi\, \mathrm{Var}_{\bar{Y}|X}(x)}} e^{-\frac{\left(y - E_{\bar{Y}|X}(x)\right)^2}{2\,\mathrm{Var}_{\bar{Y}|X}(x)}} \tag{11}$$

for $-\infty < y < +\infty$, where $p_{\bar{Y}|X}(y|x)$ is the conditional probability density of $\bar{Y}$ under given condition $X(t) = x$, and $E_{\bar{Y}|X}(x)$ and $Var_{\bar{Y}|X}(x)$ are the following conditional expectation and the conditional variance, respectively

$$\begin{aligned} E_{\bar{Y}|X}(x) &= E_{Y_i|X}(x) = E\left[g(x + \eta) \right] \\ &= x\Phi\left(\frac{x}{\sigma_\eta} \right) + \frac{\sigma_\eta}{\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_\eta^2}} \end{aligned} \tag{12}$$

$$\begin{aligned} \mathrm{Var}_{\bar{Y}|X}(x) &= \frac{1}{N}\mathrm{Var}_{Y_i|X}(x) = \frac{x^2}{N}\left(\Phi\left(\tfrac{x}{\sigma_\eta} \right) - \Phi^2\left(\tfrac{x}{\sigma_\eta} \right) \right) \\ &\quad + \frac{x\,\sigma_\eta}{\sqrt{2\pi N}} \exp\left(-\tfrac{x^2}{2\sigma_\eta^2} \right) \left(1 - 2\Phi\left(\tfrac{x}{\sigma_\eta} \right) \right) \\ &\quad + \frac{\sigma_\eta^2}{N}\left(\Phi\left(\tfrac{x}{\sigma_\eta} \right) - \tfrac{1}{2\pi} \exp\left(-\tfrac{x^2}{\sigma_\eta^2} \right) \right). \end{aligned} \tag{13}$$

In deriving (12) and (13), the nature of identical distribution and statistical independence among the conditional variables $Y_i|_{X=x}\,(1 \leq i \leq N)$ has been adopted, and then, the Fisher information can be obtained by substituting (11) into (2)–(5). Moreover, as $N \to \infty$, $\mathrm{Var}_{\bar{Y}|X}(x) \to 0$ and

$$p_{\bar{Y}|X}(y|x) \to \delta\left(y - E_{\bar{Y}|X}(x) \right). \tag{14}$$

Once the conditional probability density (10) or (11) is available, the Fisher information can be acquired by means of (4) and (5). For instance, in the limit of $N \to \infty$

$$p_{\bar{Y}}(y) = \frac{1}{\sqrt{2\pi}\,\sigma_\xi} \frac{\exp\left(-\frac{(x^* - s_a)^2}{2\sigma_\xi^2} \right)}{\Phi\left(\frac{x^*}{\sigma_\eta} \right)} \tag{15}$$

where $E_{\bar{Y}|X}(x^*) = y, y \geq 0$. Note that $E_{\bar{Y}|X}(x)$ as a function of x is continuous and monotonically increasing with $\lim_{x\to-\infty} E_{\bar{Y}|X}(x) = 0$ and $\lim_{x\to\infty} E_{\bar{Y}|X}(x) = \infty$, therefore there exists unique x^*, such that $E_{\bar{Y}|X}(x^*) = y$ for arbitrary $y > 0$. Then, the Fisher information can be acquired as

$$J_{\bar{Y}} = \left(\frac{\partial s_a(t)}{\partial a}\right)^2 \frac{1}{\sqrt{2\pi}\,\sigma_\xi} \int_{-\infty}^{+\infty} \left[\frac{(x - s_a(t))^2}{\sqrt{2\pi}\,\sigma_\xi^4}\right] dx \times \exp\left(-\frac{(x - s_a(t))^2}{2\sigma_\xi^2}\right) dx = \left(\frac{\partial s_a(t)}{\partial a}\right)^2 \frac{1}{\sigma_\xi^2}. \quad (16)$$

From here, it is clear that in the limit $N \to \infty$, the Fisher information becomes constant no matter how the nonzero intensity σ_η of the measure noise changes. This is obviously consistent with the law of large number.

The above deduction is obtained under $\sigma_\eta \neq 0$. To compare the performance of the array model in the presence of noise with that in the absence of noise, we still need to consider the case of $\sigma_\eta = 0$, the third case. Now, since all $Y_i(t)s$ are the same

$$\bar{Y}(t) = \mathrm{ReLU}(s_a(t) + \xi(t)) = (s_a(t) + \xi(t))\, I_{\{s_a(t)+\xi(t)>0\}} + 0 \cdot I_{\{s_a(t)+\xi(t)\leq 0\}}. \quad (17)$$

It is easy to see that for $y \in R$

$$P\left(\bar{Y}(t) \leq y\right) = P\left(\bar{Y}(t) = 0\right) + P\left(0 < \bar{Y}(t) \leq y\right) = F_\xi(y - s_a(t)) \cdot I_{\{y>0\}} + F_\xi(-s_a(t)) \cdot I_{\{y=0\}} \quad (18)$$

where $F_\xi(x) = (1/(2\pi)^{1/2}\sigma_\xi) \int_{-\infty}^{x} e^{-(u^2/2\sigma_\xi^2)} du$ is the accumulative distribution function of $\xi(t)$. By differentiating the both sides of (18)

$$p_{\bar{Y}}(y) = p_\xi(y - s_a(t))\; I_{\{y>0\}}(y) + F_\xi(-s_a(t))\,\delta(y) \quad (19)$$

where $p_\xi(x) = F_\xi{}'(x)$ is the density function for $\xi(t)$. Thus, in this case, the Fisher information can be attained as

$$J_{\bar{Y}} = \left(\frac{\partial s_a(t)}{\partial a}\right)^2 \left(\frac{1}{2\pi\sigma_\xi^2} \frac{1}{F_\xi(-s_a(t))} e^{-\frac{s_a(t)^2}{\sigma_\xi^2}} + \frac{1}{2\sigma_\xi^2} - \frac{s_a(t)}{\sigma_\xi^3\sqrt{2\pi}} e^{\left(-\frac{s_a(t)^2}{2\sigma_\xi^2}\right)} + \frac{1}{2\sigma_\xi^2}\,\mathrm{erf}\left(\frac{s_a(t)}{\sqrt{2}\sigma_\xi}\right)\right) \quad (20)$$

with Gaussian error function $erf(x) = (2/(\pi)^{1/2}) \int_0^x e^{-x^2} dx$.

Here, we can analytically prove that the Fisher information when $N \to \infty$ and $\sigma_\eta > 0$ is always larger than the fisher information in the absence of measurement noise. In fact, for the sake of simplicity in notion, the function $G(\cdot)$ is introduced as the difference between the Fisher information in (16) and (20), namely

$$G(s_a(t)) \triangleq \frac{1}{\sigma_\xi\sqrt{2\pi}}\, s_a(t)\, e^{-\frac{s_a^2(t)}{2\sigma_\xi^2}} - \frac{1}{2}\,\mathrm{erf}\left(\frac{s_a(t)}{\sqrt{2}\sigma_\xi}\right) - \frac{1}{2\pi F_\xi(-s_a(t))}\, e^{-\frac{s_a^2(t)}{\sigma_\xi^2}} + \frac{1}{2}. \quad (21)$$

Denote

$$G(u) = \frac{u}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} - \Phi(u) + 1 - \frac{\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\right)^2}{\Phi(-u)}, \quad (22)$$

then

$$G'(u) = -\frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} \left(u - \frac{\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\right)}{\Phi(-u)}\right)^2 \leq 0. \quad (23)$$

Equation (23) demonstrates that $G(u)$ is monotonically decreasing. Note that $G(u) \to 1$ when $u \to -\infty$, while $G(u) \to 0$ as $u \to +\infty$; hence, $G(u) \geq 0$ holds true for any u, meaning that the inequality (21) holds true for all signal $s_a(t)$. This illustrates that the Fisher information of the noisy array is really not less than the Fisher information in the deterministic counterpart.

Note that in parameter estimation, Fisher information determines the possible accuracy of estimation of a parameter from observed data. Since its inverse is the variance of an efficient estimator, when the Fisher information about the parameter is low, the accuracy of even the best estimator is poor. Thus, the above analysis illustrates that an unknown parameter can be estimated more accurate using the array of noisy ReLU activators. The measure of Fisher information has been proven powerful for quantifying SR in the array of static threshold devices [47], [52]. In virtue of these considerations, in this article, our aim is to use the noisy ReLU array in developing new intelligent fault classification algorithms, so that the benefit of noise and array can be simultaneously maximally utilized.

To expound our motivation more clearly, let us take two types of noisy analog faulty signals as input mixtures to make further explanation. The first type is the inner race faulty signal [54]

$$x_{\mathrm{inner}}(t) = \sum_i \left(1 + ae^{-d\mathrm{mod}\left(t, \frac{i}{f_s}\right)} \cos(2\pi f_r t)\right) \times \cos(2\pi f_{\mathrm{res}} t) + \xi(t) \quad (24)$$

where $a = 0.5$ is the amplitude of the fault signal, $d = -0.0488$ is the decay rate, $\mathrm{mod}(\cdot,\cdot)$ is the modular operation to keep periodicity, $f_s = 90$ is the frequency of the inner race fault signal, $f_{\mathrm{res}} = 7000$ is the resonance frequency, $f_r = 15$ is the rotating frequency, and $\xi(t)$ is the Gaussian white noise of intensity $\sigma_\xi = 0.6$. The other type is the outer race faulty signal [12]

$$x_{\mathrm{outer}}(t) = ae^{-d\mathrm{mod}\left(t, \frac{1}{f_s}\right)} \sin(2\pi f_s t) + \xi(t) \quad (25)$$

where $a = 0.5$ is the amplitude of the faulty signal, $d = 1000$ is the decay rate, $\boldsymbol{mod}(\cdot,\cdot)$ is the modular operation, $f_s = 10000$ is the frequency of the outer race faulty signal, and $\xi(t)$ is the same as above. Note that (2) only gives the Fisher information at some instant. Nevertheless, for a time varying signal, the output of two fault signals has to be observed and taken summation over a long-time interval. For the noisy periodic faulty signal, we take 100 sample instants, namely, t_j, $j = 1, \ldots, 100$ during one period time span and calculate the

one-instant Fisher information $J_{\bar{Y}_j}$ at each time instant t_j, and then, the total Fisher information is acquired as $J_{\bar{Y}} = \sum_{j=1}^{100} J_{\bar{Y}_j}$.

Whether for the noisy inner race fault signal or for the noisy outer race fault signal, as displayed in Fig. 1(c) and (d), the Fisher information exhibits a monotonically decaying dependence when the array size $N = 1$ but a nonmonotonic bell-shaped dependence on the noise intensity σ_η when $N > 1$. Moreover, this nonmonotonic dependence becomes more and more prominent as N increases before the $N \to \infty$ limit (16) is reached. The observation exactly signifies the occurrence of SSR [46]. Thus, there exists optimal noise intensity, such that the Fisher information of the collective average response of a nontrivial array ($N > 1$) attains its maximum. That is, an optimal mount of the measurement noise serves as the most appropriate impetus that can drive the input signal into a most favorable operating zone of the ReLU nonlinearity. This is exactly the principle of the novel intelligent fault diagnosis design to be presented. It is worthy to emphasize that when the limit of $N \to \infty$ is reached, as seen from (16), the Fisher information takes constant value as long as the noise intensity is larger than zero. Moreover, this constant value for $\sigma_\eta \neq 0$ and $N \to \infty$ is larger than the Fisher information (20) at $\sigma_\eta = 0$ for any finite N (as denoted by solid point on the vertical axis) and the optimal Fisher information for any finite-sized noisy ReLU array. The comparison suggests that the SSR principle recommends using an array of noisy ReLU activators as large as possible to enhance the detection of a weak signal. This is just the starting point of the present study.

III. Neural Network Design via the Large N Limit of Noisy ReLU Array

The SSR effect of the array of noisy ReLU activators is enhanced as the array size increases [47], [48], as observed from Section II. This observation ever motivated us to choose the array model of ReLU activators as nodes to propose new intelligent fault diagnosis algorithms. Nevertheless, handling an array node can drag down the network efficiency especially when the array size is large. Fortunately, in the case of $N \to \infty$ limit, the average response relation of the array has an explicit expression (12), namely

$$f(x, \sigma_\eta) \triangleq x\,\Phi\left(\frac{x}{\sigma_\eta}\right) + \frac{\sigma_\eta}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma_\eta^2}\right). \tag{26}$$

Thus, this average response limit can be chosen as the activation function. In fact, with (26) as activation function, it is equal to take the infinite-sized array of ReLU sensors as nodes of the neural networks, but no actual array simulation is involved. This can be validated by the law of large number. Note that the array response limit can display the best enhanced effect, thus this treatment should be the best choice. Moreover, the limit of array of noisy ReLU (LAN ReLU) activators (26) and its smooth derivative functions are both explicitly dependent on noise intensity, as shown in Fig. 2, and the noise dependence nature enables us to train the intensity parameter, so that the benefit of noise can be maximally utilized by back-propagation learning.

With such a perfect activation function (26) in mind, let us illustrate how to proceed the back-propagation learning in fault

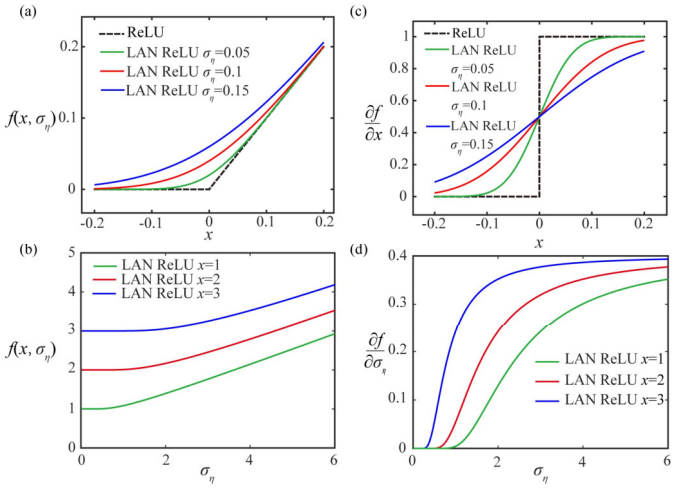


Fig. 2. Output and the derivatives of the LAN ReLU activation function. (a) Output of ReLU and LAN ReLU with respect to x. (b) Output characters of ReLU and LAN ReLU with respect to σ_η. (c) Derivative of ReLU and LAN ReLU with respect to x. (d) Derivative of ReLU and LAN ReLU with respect to σ_η.

classification. To this end, let us take a multilayer feedforward neural network of one input layer(0th layer), $L-1$ hidden layers, and one output layer(Lth layer) as example. Let N^l be the node number of the lth layer and x_{pi}^l, $p = 1, \ldots, P$, $i = 1, \ldots, N^{l-1}$ be the prepossessed input signal of the lth layer, where P denotes the total number of training samples and N^{l-1} is the length of each sample. With (26) as the lth layer's activation function $f^l(\cdot)$, the output of the lth layer reads

$$\begin{aligned} y_{pj}^l &= f^l\left(\sum_{i=1}^{N^{l-1}} x_{pi}^l w_{ij}^l + b_j^l, \sigma_\eta^l\right) \\ &= \left(\sum_{i=1}^{N^{l-1}} x_{pi}^l w_{ij}^l + b_j^l\right) \Phi\left(\frac{\sum_{i=1}^{N^{l-1}} x_{pi}^l w_{ij}^l + b_j^l}{\sigma_\eta^l}\right) \\ &\quad + \frac{\sigma_\eta^l}{\sqrt{2\pi}} \exp\left(-\frac{\left(\sum_{i=1}^{N^{l-1}} x_{pi}^l w_{ij}^l + b_j^l\right)^2}{2\left(\sigma_\eta^l\right)^2}\right) \end{aligned} \tag{27}$$

where $y_{pj}^l, p = 1, \ldots, P, j = 1, \ldots, N^l$, $w_{ij}^l, i = 1, \ldots, N^{l-1}, j = 1, \ldots, N^l$ is the weight matrix from the (l–1) th layer to the lth layer, $b_j^l, j = 1, \ldots, N^l$, where $l = 1, \ldots, L$ is the input bias for the lth layer. Here, the subscript l in the activation function and the superscript in the noise intensity both denote the lth layer.

Note that depending on the difference in components and parts where faults occur, the fault signal of rotating machinery can be divided into bearing faults and gear faults; the bearing faults mainly include inner race, outer race, and rolling ball faults [55], while the gear faults primarily consist of missing teeth, chipped teeth, and surface faults [56]. We suppose that there are $I = N^L$ fault categories, the maximal occupation proportion that determines the category of the pth input sample

Algorithm 1 Fault Classification Algorithm Based on SSR

Input: aggregate the training set $x^0 = (x^0_{pi}), i = 1, \ldots, N^0$ and the label data $z = (z_{pj}), j = 1, \ldots, I$ and initialize the weights w^l, biases b^l and noise intensity $\sigma^l_\eta, l = 1, \ldots, L$.
Output: collect the trained parameter set Θ.
For epoch $k = 0, 1, \ldots$ **do**
For the pth training sample with $p = 1, \ldots, P$ **do**
Calculate the input $(x^l_{pi}), i = 1, \ldots, N^l$ and the output (y^l_{pi}) by Eq.(27) for each layer $l = 1, \ldots, L$ and calculate y^{out}_{pi} by Eq.(28)//input forward propagation.
Calculate the loss function E using Eq.(29)//input forward propagation.
Update the output layer parameters w^L_{ij}, b^L_i and σ^L_η using Eqs.(32)- (34)//error backpropagation.
For $l = L-1, L-2, \ldots, 1$ **do**
Update the lth hidden layer parameters w^l_{ij}, b^l_i and σ^l_η using Eqs.(35)- (38)//error backpropagation
End
End
End

can be calculated by SoftMax or Gibbs activation as

$$y^{\text{out}}_{pj} = \frac{e^{\left(y^L_{pj}\right)}}{\sum_{k=1}^{I} e^{\left(y^L_{pk}\right)}}, \quad p = 1, \ldots, P, \ \ j = 1, \ldots, I. \tag{28}$$

Let $z_{pj}, p = 1, \ldots, P, j = 1, \ldots, I$ is the binary one-hot encoding matrix of the fault type [41], [57], then the loss function for this classification task can be defined by the cross entropy as $E = -(1/P)\sum_{p=1}^{P}\sum_{j=1}^{I} z_{pj} \ln y^{\text{out}}_{pj}$. Meanwhile, a penalty term given by L_2 norm of the parameter can be added to ensure the stability of the output of the neural network algorithms and to mitigate the risk of overfitting. Additionally, note that the noise intensity cannot be negative, thus we can add another penalty term. The resultant loss function reads

$$\tilde{E} = -\frac{1}{P}\sum_{p=1}^{P}\sum_{j=1}^{I} z_{pj} \ln y^{\text{out}}_{pj} + \frac{\lambda_{\text{reg}}}{M}\|\Theta\|_2^2 + \frac{\lambda_{\text{noise}}}{L}\sum_{l=1}^{L}\max\left(0, -\sigma^l_\eta\right) \tag{29}$$

where Θ is the parameter set containing weights, biases, and noise intensities of all the layers, and the total number of parameters is denoted as M. To best exploit the benefit of noise, the noise intensities on different layer are treated as different; thus, the trainable parameter set is enlarged.

We minimize the penalized loss function (29) by training the parameter set with the gradient-based backpropagation learning [58]. With the loss function being calculated in the last layer, the entire training process contains the forward propagation of input information and the backpropagation of loss functions. The involving gradient or partial derivatives associated with the activation function (26) can be calculated as follows. Note that

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x}\left(x\Phi\left(\frac{x}{\sigma_\eta}\right)\right) + \frac{\sigma_\eta}{\sqrt{2\pi}}\frac{\partial}{\partial x}\exp\left(-\frac{x^2}{2\sigma_\eta^2}\right) = \Phi\left(\frac{x}{\sigma_\eta}\right) \tag{30}$$

$$\frac{\partial f}{\partial \sigma_\eta} = x\frac{\partial}{\partial \sigma_\eta}\Phi\left(\frac{x}{\sigma_\eta}\right) + \frac{\partial}{\partial \sigma_\eta}\left(\frac{\sigma_\eta}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma_\eta^2}\right)\right) = \varphi\left(\frac{x}{\sigma_\eta}\right) \tag{31}$$

where $\Phi(r)$ and $\varphi(r)$ are the cumulative distribution and probability density of the standard normal variable, respectively. With (30) and (31) in mind, the derivatives of the penalized loss function with respect to the parameters to be trained as be calculated as follows. For the output layer

$$\frac{\partial \tilde{E}}{\partial w^L_{ij}} = \frac{2\lambda_{\text{reg}}}{M}w^L_{ij} - \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_{pj}}{\partial y^{\text{out}}_{pj}}\frac{\partial y^{\text{out}}_{pj}}{\partial y^L_{pj}}\frac{\partial y^L_{pj}}{\partial w^L_{ij}} = \frac{2\lambda_{\text{reg}}}{M}w^L_{ij} - \frac{1}{P}\sum_{p=1}^{P}\left(y^{\text{out}}_{pj} - z_{pj}\right)y^{\text{out}}_{pj}\left(1 - y^{\text{out}}_{pj}\right)x^L_{pi} \times \Phi\left(\frac{1}{\sigma^L_\eta}\left(\sum_j x^L_{pi}w^L_{ij} + b^L_j\right)\right) \tag{32}$$

$$\frac{\partial \tilde{E}}{\partial b^L_j} = \frac{2\lambda_{\text{reg}}}{M}b^L_j - \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_{pj}}{\partial y^{\text{out}}_{pj}}\frac{\partial y^{\text{out}}_{pj}}{\partial y^L_{pj}}\frac{\partial y^L_{pj}}{\partial b^L_j} = \frac{2\lambda_{\text{reg}}}{M}b^L_j - \frac{1}{P}\sum_{p=1}^{P}\left(y^{\text{out}}_{pj} - z_{pj}\right)y^{\text{out}}_{pj}\left(1 - y^{\text{out}}_{pj}\right) \times \Phi\left(\frac{1}{\sigma^L_\eta}\left(\sum_i x^L_{pi}w^L_{ij} + b^L_j\right)\right) \tag{33}$$

$$\frac{\partial \tilde{E}}{\partial \sigma^L_\eta} = \frac{2\lambda_{\text{reg}}}{M}\sigma^L_\eta - \frac{1}{P}\sum_{p=1}^{P}\sum_{j=1}^{I}\left(y^{\text{out}}_{pj} - z_{pj}\right)y^{\text{out}}_{pj}\left(1 - y^{\text{out}}_{pj}\right) \times \varphi\left(\frac{1}{\sigma^L_\eta}\left(\sum_i x^L_{pi}w^L_{ij} + b^L_j\right)\right) + \frac{\lambda_{\text{noise}}}{L}H\left(-\sigma^L_\eta\right) \tag{34}$$

where $E_{pj} = z_{pj} \ln y^{\text{out}}_{pj}$, and for the hidden layer

$$\frac{\partial \tilde{E}}{\partial w^l_{ij}} = \frac{2\lambda_{\text{reg}}}{M}w^l_{ij} - \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial y^l_{pj}}\frac{\partial y^l_{pj}}{\partial w^l_{ij}} = \frac{2\lambda_{\text{reg}}}{M}w^l_{ij} - \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial y^l_{pj}}x^l_{pi} \times \Phi\left(\frac{1}{\sigma^L_\eta}\left(\sum_i x^l_{pi}w^l_{ij} + b^l_j\right)\right) \tag{35}$$

$$\frac{\partial \tilde{E}}{\partial b^l_j} = \frac{2\lambda_{\text{reg}}}{M}b^l_j - \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial y^l_{pj}}\frac{\partial y^l_{pj}}{\partial b^l_j} = 2\lambda_{\text{reg}}b^l_j - \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial y^l_{pj}} \times \Phi\left(\frac{1}{\sigma^l_\eta}\left(\sum_i x^l_{pi}w^l_{ij} + b^l_j\right)\right) \tag{36}$$

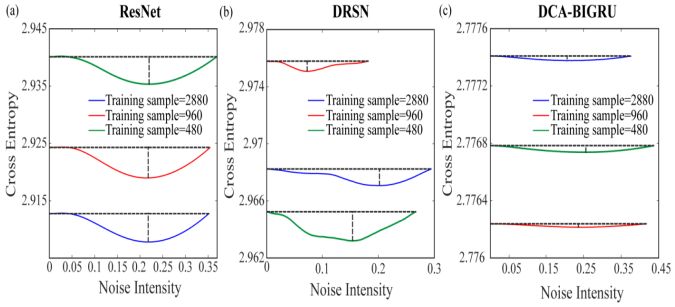


Fig. 3. Average cross entropy for the first epoch with different numbers of CWRU bearing signals under different networks. (a) ResNet, (b) DRSN, and (c) DCA-BIGRU applied LAN ReLU.

$$\frac{\partial \tilde{E}}{\partial \sigma_\eta^l} = \frac{2\lambda_{\text{reg}}}{M}\sigma_\eta^l - \frac{1}{P}\sum_{p=1}^{P}\sum_{j=1}^{N^l}\frac{\partial E_p}{\partial y_{pj}^l}\frac{\partial y_{pj}^l}{\partial \sigma_\eta^l} + \frac{\lambda_{\text{noise}}}{L}H\left(-\sigma_\eta^l\right)$$
$$= \frac{2\lambda_{\text{reg}}}{M}\sigma_\eta^l - \varphi\left(\frac{1}{\sigma_\eta^l}\left(\sum_i x_{pi}^l w_{ij}^l + b_j^l\right)\right)$$
$$\times \frac{1}{P}\sum_{p=1}^{P}\sum_{j=1}^{N^l}\frac{\partial E_p}{\partial y_{pj}^l} + \frac{\lambda_{\text{noise}}}{L}H\left(-\sigma_\eta^l\right) \quad (37)$$

$$\frac{\partial E_p}{\partial y_{pj}^l} = \sum_{m=1}^{N^{l+1}}\frac{\partial E_p}{\partial y_{pm}^{l+1}}w_{im}^{l+1}$$
$$\times \Phi\left(\frac{1}{\sigma_\eta^{l+1}}\left(\sum_{j=1}^{N^l} y_{pj}^l w_{pm}^{l+1} + b_m^{l+1}\right)\right) \quad (38)$$

where $E_p = \sum_{j=1}^{I} z_{pj} \ln y_{pj}^{\text{out}}$. For enhancing the intuitiveness, the derivative functions (30) and (31) are plotted in Fig. 2. As we know, the derivative of the ReLU activator vanishes on the negative semi-axis. But, as shown in Fig. 2(c) and (d), these shortcomings inherent in the ReLU activation are overcome by the updated activation (26). Thus, the advantage of using (26) as activation function is not limited to the only possibility of maximally utilizing noise. We will validate these merits in the next section, and our method is summarized in Algorithm 1.

IV. Numerical Verification

Three datasets, namely, one bearing dataset, one gearbox dataset, and one varying speed gearbox dataset, are used to validate the proposed algorithm. Before numerical experiments are conducted, the raw signals should be prepossessed [59], [60]. That is, for the training set, we employ a technique of moving window with overlap to increase the number of training samples from the limited original data, so that no significant bias is introduced [61]. For the test set, only moving window is applied to evade the risk of information leakage [62]. The specific window size varies with the length of different datasets and is clarified in detail in each experiment. Note that the Fisher information merely represents the amount of useful information contained in the detected data, while the cross entropy can signify the difference between the predicted probability and the real label. This is why the cross entropy provides a commonly used metric in machine learning. Here, one special point worthy to emphasize is that in the large N limit, the Fisher information (2) is independent of noise intensity, but this is no longer true for the penalized cross entropy (30), as showed in Fig. 3.

TABLE I
Network Structure of ResNet and DRSN

Layer	Parameters	Input shape	Output shape
Convolution layer	Filters=4, Kernel=3, Stride=1	(1, 1, 1026)	(1, 4, 1024)
ReLU / LAN-ReLU	—	(1, 4, 1024)	(1, 4, 1024)
RBU1 / RSBU1	Filters=4, Stride=1	(1, 4, 1024)	(1, 4, 1024)
ReLU / LAN-ReLU	—	(1, 4, 1024)	(1, 4, 1024)
RBU2 / RSBU2	Filters=8, Stride=2	(1, 4, 1024)	(1, 8, 512)
ReLU / LAN-ReLU	—	(1, 8, 512)	(1, 8, 512)
RBU3 / RSBU3	Filters=16, Stride=2	(1, 8, 512)	(1, 16, 256)
ReLU / LAN-ReLU	—	(1, 16, 256)	(1, 16, 256)
Average-pooling layer	Kernel size=4	(1, 16, 256)	(1, 16, 64)
Flatten layer	—	(1, 16, 64)	(1, 1024)
Fully connected layer	Neuron number=classes	(1, 1024)	(1, classes)
SoftMax	—	(1, classes)	(1, classes)

TABLE II
Network Structure of DCA-BIGRU

Layer	Parameters	Input shape	Output shape
Convolution layer1	Filters=50, Kernel=18, Stride=2	(1,1,1024)	(1,50,504)
ReLU / LAN-ReLU	—	(1,50,504)	(1,50,504)
Convolution layer2	Filters=30, Kernel=10, Stride=2	(1,50,504)	(1,30,248)
ReLU / LAN-ReLU	—	(1,30,248)	(1,30,248)
Max-pooling	Kernel=2	(1,30,248)	(1,30,124)
Convolution layer3	Filters=50, Kernel=6, Stride=1	(1,1,1024)	(1,50,1019)
ReLU / LAN-ReLU	—	(1,50,1019)	(1,50,1019)
Convolution layer4	Filters=40, Kernel=6, Stride=1	(1,50,1019)	(1,40,1014)
ReLU / LAN-ReLU	—	(1,40,1014)	(1,40,1014)
Max-pooling	Kernel=2	(1,40,1014)	(1,40,507)
Convolution layer5	Filters=30, Kernel=6, Stride=1	(1,40,507)	(1,30,502)
ReLU / LAN-ReLU	—	(1,30,502)	(1,30,502)
Convolution layer6	Filters=30, Kernel=6, Stride=2	(1,30,502)	(1,30,249)
Max-pooling	Kernel=2	(1,30,249)	(1,30,124)
Attention	Kernel=1, Stride=1	(1,30,124)	(1,30,124)
GRU	Units=128	(1,30,124)	(1,30,128)
Adaptive average pooling	—	(1,30,128)	(1,30,1)
Fully connected layer	Neuron number=classes	(1,30,1)	(1,classes)
SoftMax	—	(1,classes)	(1,classes)

A. Network Architecture and Parameter Initialization

In machine learning, proper architectures for a specific task are always important for achieving high accuracy and less energy cost [63], [64]. In this article, ResNet [65], deep

Algorithm 2 Initialization of Noise Intensity in LAN ReLU

Input: aggregate the training set $x^0 = (x_{pi}^0), i = 1, .., N^0$, the label data $z = (z_{pj}), j = 1, \ldots, I$, initialize the weights w^l, biases b^l and a vector of noise intensity $\sigma_\eta = (\sigma_{\eta m}), m = 1, \ldots, M$.

Output: the optimal noise intensity $\sigma_{\eta m^*}$.

For the mth noise intensity $m = 1, \ldots, M$ **do**

For the pth training sample with $p = 1, \ldots, P$ **do**

For each layer $l = 1, \ldots, L$, initialize the noise intensity $\sigma_\eta^l = \sigma_{\eta m}$. Calculate the input $(x_{pi}^l), i = 1, \ldots, N^l$, the output (y_{pi}^l) by Eq.(26) for each layer and calculate y_{pi}^{out} by Eq.(27).

End

Calculate the loss function $\tilde{E}$ using Eq.(28) and denote as $\tilde{E}_m$.

End

Find $m^* = \arg\min_m \tilde{E}_m$ and output $\sigma_{\eta m^*}$ as the initial value of LAN-ReLU.

residual shrinkage network (DRSN) [25], [66], and a smartly designed network with attention mechanism (DCA-BIGRU) [24], as shown in Tables I and II, are employed as backbone to develop new algorithm. In Table I, the RBU represents the residual block unit, where the input of the block is directly fed to the output layer, allowing the network to learn residual functions with reference to the input signal [65], while the RSBU represents residual shrinkage building unit to filter out unimportant features [25]. In Table II, the DCA-BIGRU [24] introduces a similar attention block as RBU and RSBU.

The weights and biases of all the networks are randomly initialized. Note that the noise intensity at each layer is not treated as hyperparameter but trainable, thus it also needs initialization. To this end, for each noise-intensity candidate, we perform a single forward pass and compute the cross-entropy loss, and the candidate with the lowest loss is then selected as the initial value. It should be mentioned that this technique is often used for selecting initial hyperparameter values [57] and also used in all the numerical experiments of this article if no special demonstration is provided. As shown in Fig. 3, the cross entropy as a function of noise intensity has minimum for all the network structures under our study. The smaller the cross entropy, the accuracy of the achieved classification. We assign all the involving noise intensities with a uniform initial value in the same network, where the cross entropy attains its minimum, as summarized in Algorithm 2. Here, we emphasize that within a $\pm 50\%$ range around the initial noise values determined by Algorithm 2, the test accuracy remains relatively stable, indicating that the algorithm is not sensitive to the initial noise intensity. Beyond this range, test accuracy may decline, particularly under limited training samples, but still outperforms that of ReLU. Therefore, unless otherwise specified, we do not further discuss the selection of initial noise values or their impact. As all the preparations get ready, the ADAM optimizer [67], which can usually achieve faster training speed, is adopted to update the network parameters. For all the datasets in Section IV, the learning rates for ResNet and DRSN are set to 0.004. For DCA-BIGRU, the learning rate is set to 0.004 for the CWRU dataset in Section IV-A, 0.0001 for the SEU dataset in Section IV-B, and 0.001 for the HUST dataset in Section IV-C.

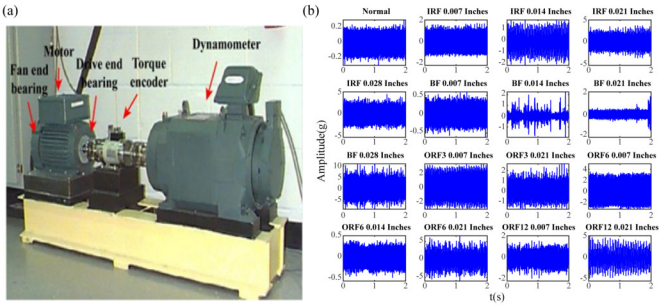
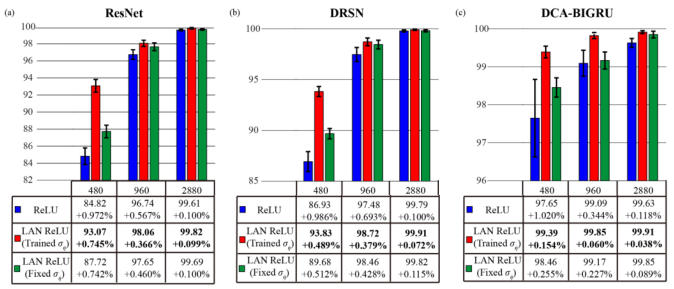


Fig. 4. Overview of the apparatus (a) and signals (b) in CWRU bearing dataset.

B. CWRU Bearing Dataset

1) Dataset Description: The CWRU bearing dataset is a widely used benchmark dataset in machinery fault diagnosis. The experimental platform comprises a motor, torque encoder, and dynamometer. Data from the drive-end (DE) bearing housing are sampled at 12 and 48 kHz, while supplementary data from the fan-end (FE) bearing housing are exclusively sampled at 12 kHz. The dataset covers various fault types, including inner race fault (IRF), ball fault (BF), and outer race fault (ORF), with diameters 0.007, 0.014, 0.021, and 0.028 inches, alongside normal data. The ORF can be further distinguished via position into three categories: 3 o'clock, 6 o'clock, and 12 o'clock, namely, ORF 3, 6, and 12. Particularly, there is no 0.014-in fault in the 3 o'clock and 12 o'clock categories, and there is no 0.028-in fault in the 3 o'clock, 6 o'clock, or 12 o'clock either. All data are collected in four load conditions: 0, 1, 2 and 3 Horse powers (Hp), simulating diverse operational stresses. In this article, we utilize the 12-kHz DE bearing data with load from 0 to 2 Hp, and we only distinguish differences in fault diameter and location, regardless of varying loads. Thus, the dataset comprises of 16 data categories in total, as shown in Fig. 4(b). The training samples are derived by a preprocessing window of length 1024 and 60 being the overlapping size, while the test samples are processed with the same window size but without overlap. For each category, the whole dataset comprises 180 training samples and 60 testing samples.

2) Test Accuracy Under Different Training Sample Sizes: We evaluate the performance of the new algorithms under varying training sample sizes. As we known, a limited training dataset can affect the model's generalization ability by causing overfitting or suboptimum [68], [69], while in the field of fault diagnosis, the situation of limited labeled data is ubiquitous. Thus, a good intelligent fault diagnosis algorithm must be tested under different numbers of training samples. In this study, we define "limited sample" based on the ratio of training to testing samples. According to [24], when the proportion of training samples per class is lower than 50%, it can be considered as limited samples. We fix the test set with 960



	ResNet 480	ResNet 960	ResNet 2880
ReLU	84.82 +0.972%	96.74 +0.567%	99.61 +0.100%
LAN ReLU (Trained σ_η)	**93.07 +0.745%**	**98.06 +0.366%**	**99.82 +0.099%**
LAN ReLU (Fixed σ_η)	87.72 +0.742%	97.65 +0.460%	99.69 +0.100%

	DRSN 480	DRSN 960	DRSN 2880
ReLU	86.93 +0.986%	97.48 +0.693%	99.79 +0.100%
LAN ReLU (Trained σ_η)	**93.83 +0.489%**	**98.72 +0.379%**	**99.91 +0.072%**
LAN ReLU (Fixed σ_η)	89.68 +0.512%	98.46 +0.428%	99.82 +0.115%

	DCA-BIGRU 480	DCA-BIGRU 960	DCA-BIGRU 2880
ReLU	97.65 +1.020%	99.09 +0.344%	99.63 +0.118%
LAN ReLU (Trained σ_η)	**99.39 +0.154%**	**99.85 +0.060%**	**99.91 +0.038%**
LAN ReLU (Fixed σ_η)	98.46 +0.255%	99.17 +0.227%	99.85 +0.089%

Fig. 5. Classification accuracy on test set with 2880, 960, and 480 training samples on the (a) ResNet, (b) DRSN, and (c) DCA-BIGRU. Our method with trained noise intensity σ_η (red bar) is compared with the fixed σ_η case (green bar) and ReLU (blue bar).

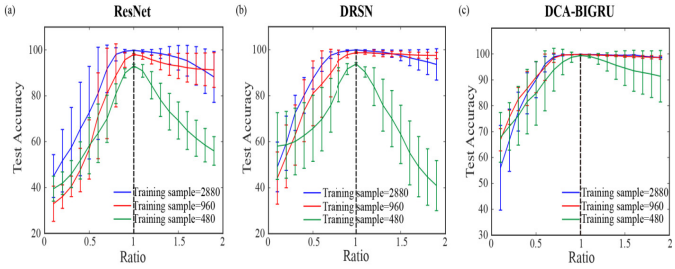


Fig. 6. Test accuracy versus noise intensity scaling ratio for (a) ResNet, (b) DRSN, and (c) DCA BIGRU under 2880 (blue curve), 960 (red curve), and 480 (green curve) training samples on CWRU dataset.

test samples but randomly choose 960 and 480 samples from the entire training set as the training sets, which changes the ratio of training to test samples from 1:1 to 1:2, receptively. We also use the entire training set to train the networks. As shown in Fig. 5(a), when trained on the ResNet, the proposed method achieves the highest accuracy of 99.82%, 98.06%, and 93.07% under all the training sample sizes, surpassing the test accuracy of the ResNet in the absence of noise and in the presence of fixed noise. Here, by fixed noise, it means that the noise intensity is fixed to its initialized value. As shown in Fig. 5(b), when trained on DRSN, the proposed method again attains the highest test accuracy of 99.91%, 98.72%, and 93.93% under various training sample sizes among all the three cases and the test accuracy in the case of fixed noise is higher than that is the case of without noise. Similarly, on DCA-BIGRU, applying LAN ReLU achieves test accuracies of 99.91%, 99.85%, and 99.39% across various training sample sizes, outperforming both ReLU and fixed noise conditions [Fig. 5(c)]. From all the observations with this bearing dataset, it can be concluded that injection of noise with initialized value (as shown by Fig. 3) than no noise injection and injection of noise with trained intensity can achieve the best performance. In particular, it can be observed from Fig. 5 that our method can achieve more significant improvement in test accuracy under small sample conditions, indicating a greater suitability for training on limited datasets.

3) Effect of Stochastic Resonance in the Neural Network: In Section II, we have demonstrated the phenomenon of SSR in the array of noisy ReLU activators, which acts as the foundation of constructing new activation function. Now,

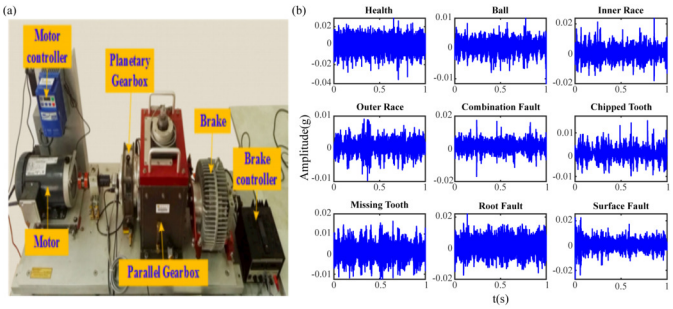


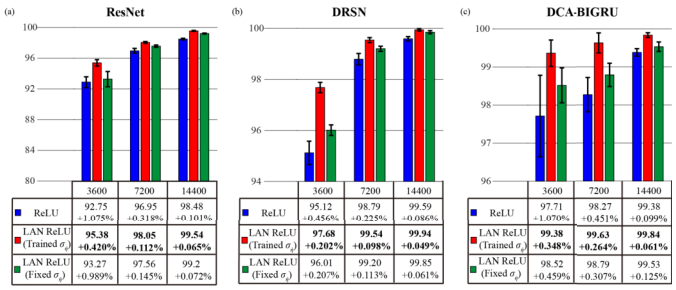
Fig. 7. Overview of the apparatus (a) signals and (b) SEU gearbox dataset.

let us examine whether SR occurs in "trained" models by observing the dependence of the test accuracy on the scaled noise intensity under different sample sizes and network architectures. Here, the "trained" model means the model with trained weight parameters but tunable noise intensity, while the scaled noise intensity is referred to as the ratio of noise intensity over the trained intensity. As seen in Fig. 6(a), the test accuracy on ResNet always decreases as the scaled noise intensity deviates from one, signifying that the trained noise intensity is optimal. Similar observations are found on DRSN and DCA-BIGRU as well. Therefore, with the test accuracy as quantifying index, the phenomenon of SR can indeed occur in real machine learning framework.

C. SEU Gearbox Dataset

1) Dataset Description: The dataset of gearbox vibration signals provided by Southeast University (SEU) consists of a bearing dataset and a gear dataset [56]. It contains two kinds of speed and loading conditions: 20 Hz–0 V and 30 Hz–2 V. Gears have four types of faulty signals, namely, chipped tooth, missing tooth, root fault, and surface fault; bears also have four types of faulty signals, namely, inner race fault, outer race fault, combination fault of inner race and outer race, and ball fault. Note that the healthy statuses in the two working conditions can be mixed into one healthy status; thus, this dataset comprises a total of nine data categories, as shown in Fig. 7(b). The training samples are generated by a window of 1024 points shifting with 512 points, while the test samples are generated using the same window size and without overlap. The whole dataset consists of 1600 training samples and 800 test samples for each category.

2) Test Accuracy Under Different Training Sample Sizes: To demonstrate the adaptability of the new algorithms based on ResNet and DRSN in the ubiquitous limited training datasets, the relevant performance should be examined under varying training sample sizes. We use the whole training set to train the both networks, and we also randomly select 7200 and 3600 samples from the entire training set as the training sets but keep the number of test samples fixed at 7200. As shown in Fig. 8(a), when trained on ResNet, the proposed algorithm achieves the accuracy of 99.54%, 98.05%, and 95.38%, respectively, surpassing the test accuracy of ResNet without noise and fixed noise intensity at the initial value. As shown in Fig. 8(b), when trained on DRSN, the



	3600	7200	14400
ReLU	92.75 +1.075%	96.95 +0.318%	98.48 +0.101%
LAN ReLU (Trained σ_η)	**95.38 +0.420%**	**98.05 +0.112%**	**99.54 +0.065%**
LAN ReLU (Fixed σ_η)	93.27 +0.989%	97.56 +0.145%	99.2 +0.072%

	3600	7200	14400
ReLU	95.12 +0.456%	98.79 +0.225%	99.59 +0.086%
LAN ReLU (Trained σ_η)	**97.68 +0.202%**	**99.54 +0.098%**	**99.94 +0.049%**
LAN ReLU (Fixed σ_η)	96.01 +0.207%	99.20 +0.113%	99.85 +0.061%

	3600	7200	14400
ReLU	97.71 +1.070%	98.27 +0.451%	99.38 +0.099%
LAN ReLU (Trained σ_η)	**99.38 +0.348%**	**99.63 +0.264%**	**99.84 +0.061%**
LAN ReLU (Fixed σ_η)	98.52 +0.459%	98.79 +0.307%	99.53 +0.125%

Fig. 8. Classification accuracy on test set with 3600, 7200, and 14 400 training samples on the (a) ResNet, (b) DRSN, and (c) DCA-BIGRU. Our method with trained noise intensity σ_η (red bar) is compared with the fixed σ_η case (green bar) and the ReLU (blue bar).

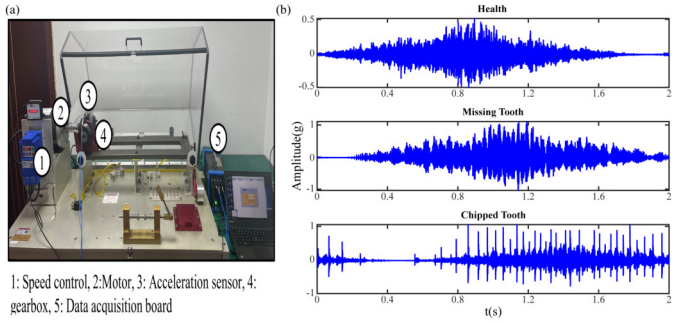
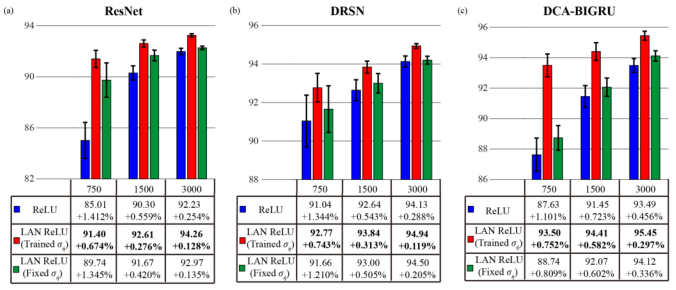


Fig. 9. Overview of the apparatus (a) and signals (b) in HUST gearbox dataset.

test accuracy of the proposed algorithm achieves the highest accuracy of 99.94%, 99.54%, and 97.68% among the cases of trained noise, fixed noise, and without noise. Similarly, when trained on DCA-BIGRU with LAN ReLU, the test accuracies reach 99.84%, 99.63%, and 99.38%, all higher than those with ReLU and fixed noise intensity [Fig. 8(c)]. It is clearly known from the observation with the gearbox dataset that the both algorithms have good feasibility in treating this complex diagnosis task under small sample conditions, and particularly, the smaller the training sample size, the more evident the relevant improvement.

D. HUST Gearbox Dataset

1) Dataset Description: The Huazhong University of Science and Technology (HUST) gearbox dataset provides vibration signals from gearbox in three different states (health, missed tooth, and chipped tooth) under 30 distinct working conditions (five types of loads and six types of speed) [70]. The working loads include 0, 0.113, 0.226, 0.339, and 0.452 Nm. The working speed covers 20, 25, 30, 35, and 40 Hz and a varying speed 0–40–0 Hz. In this dataset, the performance of our algorithm is evaluated under five working loads at the challenging varying speed of 0–40–0 Hz. Note that the missing tooth faulty signals under different loads are mixed into one category of missing tooth faulty signal. Similarly, the chipped tooth faulty signals are mingled into one category of chipped tooth faulty signal, and the healthy signals under different working conditions are blended into one healthy signal. Thus, there are three kinds of fault categories in total, as shown in



	750	1500	3000
ReLU	85.01 +1.412%	90.30 +0.559%	92.23 +0.254%
LAN ReLU (Trained σ_η)	**91.40 +0.674%**	**92.61 +0.276%**	**94.26 +0.128%**
LAN ReLU (Fixed σ_η)	89.74 +1.345%	91.67 +0.420%	92.97 +0.135%

	750	1500	3000
ReLU	91.04 +1.344%	92.64 +0.543%	94.13 +0.288%
LAN ReLU (Trained σ_η)	**92.77 +0.743%**	**93.84 +0.313%**	**94.94 +0.119%**
LAN ReLU (Fixed σ_η)	91.66 +1.210%	93.00 +0.505%	94.50 +0.205%

	750	1500	3000
ReLU	87.63 +1.101%	91.45 +0.723%	93.49 +0.456%
LAN ReLU (Trained σ_η)	**93.50 +0.752%**	**94.41 +0.582%**	**95.45 +0.297%**
LAN ReLU (Fixed σ_η)	88.74 +0.809%	92.07 +0.602%	94.12 +0.336%

Fig. 10. Classification accuracy on test set with 3000, 1500, and 750 training samples on the (a) ResNet, (b) DRSN, and (c) DCA-BIGRU. Our method with trained noise intensity σ_η (red bar) is compared with the fixed σ_η case (green bar) and the ReLU (blue bar).

Fig. 9(b). The training samples are generated using a window of 1024 points including 80 times of overlapping, while the test samples are created with the same window size but without overlap.

2) Test Accuracy Under Different Training Sample Sizes: We use the whole training set to train the new algorithms, and we also randomly choose 1500 and 750 samples from the whole training set to simulate the limited sample cases as we have done with the CWRU dataset and the SEU dataset. As shown in Fig. 10(a), when trained on the ResNet, our method achieves a test accuracy of 94.26%, 92.61%, and 91.4% for the training sample of size 3000, 1500, and 750, respectively, surpassing the according accuracy of the without noise case and the fixed noise case. Moreover, when trained on DRSN, our method achieves a test accuracy of 94.94%, 93.84%, and 92.77%, also higher than the according accuracy obtained without noise and with fixed noise, as displayed in Fig. 10(b). When trained on DCA-BIGRU with LAN-ReLU, the test accuracies are 95.45%, 94.41%, and 93.50% for training sample sizes of 3000, 1500, and 750, respectively, all higher than those obtained with ReLU and fixed noise intensity [Fig. 10(c)]. The observations demonstrate that the algorithm proposed in this article performs well with this complex speed varying dataset. Once again, it can be seen that noise plays a better improvement role when the training sample size is limited.

E. Computational Complexity of LAN ReLU and Comparison With Other Methods

Besides test accuracy, memory usage, floating point operations (FLOPs), and test time are also critical indexes that can evaluate the overall performance of neural network algorithms. As usual, the memory usage is defined as the peak GPU memory consumed during a single forward pass, which reflects the model's resource footprint and its scalability on devices with limited memory, while FLOPs quantify the total number of arithmetic operations per forward pass, serving as a hardware agnostic indicator of theoretical computational complexity. Here, the computational complexity is conducted on a workstation with an RTX 4090 GPU and an Intel[1] Core[2]

[1]Registered trademark.
[2]Trademarked.

TABLE III
COMPUTATIONAL COMPLEXITY FOR LAN RELU

Network	Activation	Dataset	Test time	Memory usage	FLOPs
ResNet	ReLU	CWRU	0.05 s	20.70 MB	1.54 M
		SEU	0.15 s		
		HUST	0.05 s		
	LAN ReLU	CWRU	0.09 s	20.73 MB	2.35 M
		SEU	0.39 s		
		HUST	0.07 s		
DRSN	ReLU	CWRU	0.09 s	26.76 MB	1.71 M
		SEU	0.29 s		
		HUST	0.07 s		
	LAN ReLU	CWRU	0.14 s	26.80 MB	2.46 M
		SEU	0.79 s		
		HUST	0.13 s		
DCA-BIGRU	ReLU	CWRU	0.04 s	75.58 MB	24.61 M
		SEU	0.13 s		
		HUST	0.04 s		
	LAN ReLU	CWRU	0.06 s	115.2 MB	26.67 M
		SEU	0.26 s		
		HUST	0.06 s		

i9-14900K 3.20-GHz CPU. As shown in Table III, on ResNet, the LAN ReLU method increases the test time from 0.05 to 0.09 s on CWRU, from 0.15 to 0.39 s on SEU, and from 0.05 to 0.07 s on HUST, while the peak memory remains virtually unchanged and the FLOPs rise from 1.54 to 2.35 M, when compared with the ReLU method. The negligible change in peak memory usage shows that resource consumption remains dominated by the model architecture, while the modest increases in the test time and FLOPs approximately fall within acceptable limits, demonstrating that LAN ReLU can enhance performance without compromising computational efficiency. On the DRSN architecture, the performance of the LAN ReLU method is similar to its performance on ResNet. Furthermore, on more complex DCA-BIGRU architecture, the increase in computation time introduced by the LAN ReLU method remains small, and relative to the overall FLOPs with the ReLU method, the approximately 10% rise in FLOPs is negligible. The vertical but comprehensive comparison demonstrates that the LAN ReLU method should be an efficient and potential method in the field of intelligent fault diagnosis.

In the previous subsections, we have compared the test accuracy of the LAN ReLU method with the ReLU method on three typical backbones. Here, let us further evaluate the LAN ReLU method against other state-of-the-art networks, including models designed for limited-sample learning [24], [68], a recent advanced technique based on ensemble learning [71], transformer [72], and temporal convolutional network [73], across the CWRU, SEU, and HUST datasets. The transformer and TCN models are configured with slightly more parameters than DCA-BIGRU, to ensure a fair comparison. As shown in Table IV, on the CWRU dataset with 2880 samples, our method attains a peak accuracy of 99.91%, while under low-sample conditions, it maintains 99.85% and 99.39% accuracy. Notably, the method exhibits minimal performance variance, demonstrating exceptional robustness. Moreover, on more complex gearbox datasets, whether under constant-speed conditions (SEU) or variable-speed operations (HUST), our approach consistently achieves the highest diagnostic precision. Although the standard deviation of our method is slightly higher than some of those baseline methods, it can also be lower than some of those recent methods. This horizontal comparison further demonstrates the superiority of the LAN ReLU method and we believe these methods can be further improved by combining LAN ReLU.

In addition, we evaluated our method against other activation functions, including Swish [74] and GELU [75], as well as noise-based techniques, such as dropout [76] and NoisyNet [77], by fixing the network architecture to ResNet. As shown in Table V, our LAN ReLU method consistently outperforms other mainstream activation functions, such as Swish and GELU, as well as noise-based techniques, including dropout and NoisyNet, across all three datasets and varying sample sizes. Furthermore, the results show that LAN ReLU is not only effective as a standalone approach but also compatible with other noisy techniques. When combined with dropout, it yields higher accuracy, indicating its potential as a flexible component in model design.

V. NOISE ROBUSTNESS OF OUR ALGORITHM

Note that the vibration signals of rotating machines collected from real world are inevitably contaminated by complex working conditions. Therefore, it is necessary to examine the robustness of the proposed method against noise. To do so, we add Gaussian white noise into the datasets, such that the contaminated input has prescribed SNR (the definition can be found in [79]) and conduct numerical experiments using the same architectures. In this section, we adopt the full training sets described in Section IV for all three datasets. The learning rate and optimizer remain identical to those in Section IV, while the noise intensity parameters are initialized following Algorithm 2.

A. CWRU Bearing Dataset

First, we consider the case where the noise was added in the first way. For the CWRU dataset, when the SNR of the noise contaminated input is 8, 4, 0, and −4 dB in order, the test accuracy of our algorithm trained on the ResNet is 98.91%, 98.30%, 96.05%, and 90.54%, respectively, as shown in Fig. 11(a). Clearly, the test accuracy achieved by the proposed algorithm surpasses the accuracy obtained in the absence of noise and the accuracy obtained with fixed noise. The same tendency can be observed from the training of our algorithms on DRSN and DCA-BIGRU, as shown in Fig. 11(b) and (c). Thus, it is natural to conclude that the robustness of the proposed algorithms with trained intensity of noise can outperform the robustness of the according algorithms without noise or with fixed noise. The advantage of the proposed algorithms should be attributed to the training of the noise intensity of activation function (26). In fact, the trained noise intensities of the three network architectures are shown in Fig. 11(d)–(f). It is easy to see that the noise intensity at each layer is trained to a higher value enabling the network to better withstand noisy conditions as the SNR

TABLE IV
COMPARISON WITH OTHER METHODS IN TEST ACCURACY

Method	CWRU			SEU			HUST		
	2880	**960**	**480**	**14400**	**7200**	**3600**	**3000**	**1500**	**750**
Ref.[68]	99.14%	96.88%	92.13%	97.54%	95.56%	84.55%	91.87%	90.31%	86.93%
	+0.100	+0.071	+0.917	+0.250	+0.326	+1.475	+0.409	+0.600	+0.705
Ref.[61]	98.99%	98.32%	93.19%	99.50%	97.44%	95.02%	89.55%	86.98%	84.01%
	+0.170	+0.484	+1.056	+0.178	+0.741	+1.743	+0.402	+0.389	+0.886
Ref.[71]	99.83%	99.62%	99.00%	99.91%	99.52%	99.36%	94.86%	92.85%	89.03%
	+0.095	+0.064	+0.289	+0.025	+0.095	+0.287	+0.208	+0.378	+2.113
Ref.[24]	99.78%	99.48%	97.97%	99.28%	98.41%	98.13%	94.88%	92.74%	87.82%
	+0.086	+0.531	+0.927	+0.296	+0.959	+1.070	+0.395	+0.760	+2.032
Ref.[78]	99.27%	98.06%	94.37%	99.18%	97.66%	93.11%	93.38%	86.75%	65.31%
	+0.096	+0.415	+1.305	+0.025	+0.446	+0.497	+0.408	+0.872	+0.430
Ref.[72]	99.75%	98.41%	96.46%	99.66%	99.39%	96.65%	91.94%	88.51%	84.69%
	+0.100	+0.053	+0.429	+0.078	+0.093	+0.455	+0.389	+0.470	+0.759
Ref.[73]	99.81%	99.17%	98.41%	99.77%	99.54%	99.37%	94.60%	93.92%	90.20%
	+0.131	+0.225	+0.881	+0.193	+0.156	+0.227	+0.556	+0.567	+0.744
Ours	**99.91%**	**99.85%**	**99.39%**	**99.94%**	**99.63%**	**99.38%**	**95.45%**	**94.41%**	**93.50%**
	+0.038	**+0.060**	**+0.154**	**+0.049**	**+0.264**	**+0.348**	**+0.297**	**+0.582**	**+0.752**

TABLE V
COMPARISON WITH OTHER ACTIVATION FUNCTIONS AND NOISE-BASED TECHNIQUES IN TEST ACCURACY

Method	CWRU			SEU			HUST		
	2880	**960**	**480**	**14400**	**7200**	**3600**	**3000**	**1500**	**750**
Swish[74]	99.54%	94.74%	82.89%	99.51%	97.86%	93.05%	91.31%	88.83%	84.85%
	+0.104	+0.467	+1.327	+0.044	+0.150	+0.283	+0.210	+0.715	+1.296
GELU[75]	99.67%	95.37%	84.56%	99.47%	97.91%	92.65%	91.52%	90.35%	86.08%
	+0.126	+0.486	+1.877	+0.091	+0.070	+0.686	+0.282	+0.373	+1.453
NoisyNet[77]	99.53%	95.44%	86.67%	99.45%	96.22%	88.79%	92.58%	91.29%	88.49%
	+0.156	+0.682	+1.211	+0.146	+1.068	+1.516	+0.101	+0.075	+1.937
Dropout[76]	99.79%	98.00%	93.02%	99.41%	97.67%	91.32%	93.78%	92.36%	88.89%
	+0.127	+0.077	+0.435	+0.118	+0.664	+0.789	+0.104	+0.287	+0.429
Ours	**99.82%**	**98.06%**	**93.07%**	**99.54%**	**98.05%**	**95.38%**	**94.26%**	**92.61%**	**91.40%**
	+0.099	**+0.567**	**+0.745**	**+0.065**	**+0.112**	**+0.420**	**+0.128**	**+0.276**	**+0.674**
Ours + Dropout	**99.90%**	**99.13%**	**94.79%**	**99.75%**	**99.39%**	**98.20%**	**94.98%**	**93.93%**	**93.02%**
	+0.091	**+0.577**	**+0.185**	**+0.076**	**+0.195**	**+0.415**	**+0.204**	**+0.241**	**+0.474**

of the input signal increases. Note that the optimal values for independent noise intensities are all not zero and away from zero, thus we assert that training the intensity of noise plays a critical role in enabling our algorithms to adapt to noisy inputs. Moreover, for ResNet and DRSN, the trained noise intensity decreases with depth because shallow convolutional layers benefit from stronger noise to enhance low-level feature detection, while deep layers require lower noise to maintain stable high-level representations [43]. In contrast, in DCA-BIGRU, the trained noise intensity increases with depth, with shallow GRU layers having low noise to maintain gate sensitivity to detailed features and deeper layers having higher noise to reduce reliance on single-step information to improve long-term generalization.

B. SEU Gearbox Dataset

For the SEU dataset, we only consider the case where the noise is added in the first way. When the robustness of the proposed method is tested on ResNet, as shown in Fig. 12(a), as the SNR of the contaminated dataset changes from 8 and 4 to 0 dB and −4 dB, the test accuracy attains 98.92%, 95.49%, 85.51%, and 69.65%, respectively. Moreover, the test accuracy of the proposed method on ResNet is markedly higher than the test accuracy of the counterpart in the absence of noise or in the presence of noise with fixed intensity. When the robustness of the proposed method is tested on DSRN, as shown in Fig. 12(b), as the SNR of the contaminated dataset varies from 8 and 4 dB to 0 and −4 dB, the test accuracy attains 99.74%, 99.71%, 89.69%, and 73.31%, respectively. Again, the performance of the proposed method in noisy environment is found superior to the performance of the counterpart without noise or with fixed noise. When the robustness of the proposed method is tested on DCA-BIGRU, as the SNR of the contaminated dataset varies from 8 to 4, 0, and −4 dB, the test accuracies reach 99.43%, 96.54%, 89.06%, and 76.89%, respectively [Fig. 12(c)]. This clearly illustrates the strong adaptability of the proposed method and the superiority of the proposed

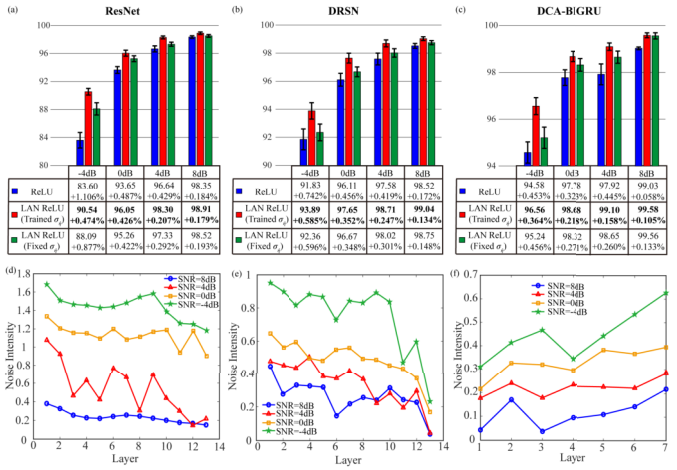


Fig. 11. Classification accuracy on test set for (a) ResNet, (b) DRSN, and (c) DCA-BIGRU with 2880 training samples when the original signals are under 8-, 4-, 0-, and −4-dB environments, along with the optimal noise intensity for (d) ResNet, (e) DRSN, and (f) DCA-BIGRU.

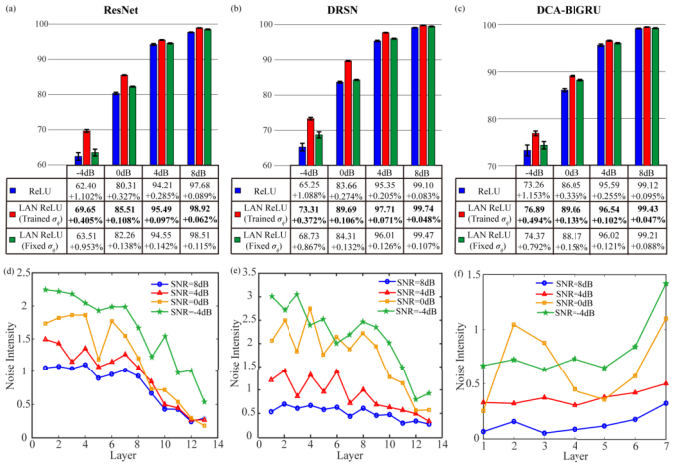


Fig. 12. Classification accuracy on the test set for (a) ResNet, (b) DRSN, and (c) DCA-BIGRU with 14 400 training samples under 8, 4, 0, and −4 dB environments, along with the optimal noise intensity for (d) ResNet, (e) DRSN, and (f) DCA-BIGRU.

method becomes more prominent in accuracy improvement when the dataset is badly contaminated. Moreover, as seen from Fig. 12(d)–(f), the trained noise intensity across layers exhibits a similar trend to that observed on the CWRU dataset, and the optimal noise intensity still converges to a higher value enabling the network to better withstand noisy conditions as the SNR of the input signal increases.

C. HUST Gearbox Dataset

For the HUST gearbox dataset, we continue to consider the case where the noise is only added in the first way. When noise is simultaneously added to the training set and the test set, as the noise level changes from 8, 4, and 0 dB to −4 dB in order, the test accuracies of our method on ResNet are 89.2%, 88.64%, 86.4%, and 80.46%, respectively, as seen from Fig. 13(a). It is not hard to see that the performance of the proposed method outperforms the performance of the counterpart without noise or with fixed noise. As the noise

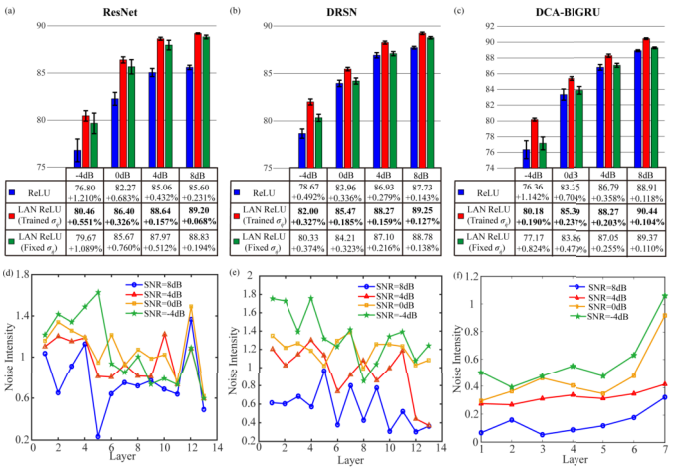


Fig. 13. Classification accuracy on the test set for (a) ResNet, (b) DRSN, and (c) DCA-BIGRU with 3000 training samples for HUST dataset under 8-, 4-, 0-, and −4-dB environments, along with the optimal noise intensity for (d) ResNet, (e) DRSN, and (f) DCA-BIGRU.

level changes similarly, the test accuracy of our method on DRSN is 89.25%, 88.27%, 85.47%, and 82%, respectively, as shown in Fig. 13(b). For DCA-BIGRU, our method achieves test accuracies of 90.44%, 88.27%, 85.39%, and 80.18% at SNRs of 8, 4, 0, and −4 dB, respectively, consistently surpassing the results obtained without noise or with fixed noise [Fig. Fig. 13(c)]. Again, it can be seen that the proposed method has better accuracy than the accuracy of the counterpart without noise or with fixed noise. The nonvanishing intensity of noise in Fig. 13(d)–(f) suggests its nontrivial role in this fault classification task and the trend of trained noise intensity across layers is consistent with that observed in the previous datasets.

VI. Conclusion and Discussion

We presented a novel intelligent fault diagnosis algorithm based on the principle of SSR in the array of noisy ReLU sensors. The Fisher information was adopted as a quantifying index to display this counterintuitive phenomenon, and then, the noise-dependent average input-output relation limit of the array is taken as activation function. By training the noise intensity independently, the maximal utilization of noise has been achieved in the proposed algorithm. It has been verified that the proposed method, namely, the LAN ReLU method, can obviously improve the diagnostic accuracy, particularly when the size of the training samples is limited. Furthermore, it has been found that the LAN ReLU method has strong robustness to noise, revealing its potential for real-world applications where data collecting often suffers from inevitable interference. Although the present study has verified the superiority of the LAN ReLU method on publicly available benchmark datasets under controlled conditions, we look forward to future work validating its effectiveness in real-world scenarios from industrial deployment.

It should be emphasized that the significance of the present study is far more than proposing an updated intelligent fault diagnosis method. It actually tells us a direction to make

efforts toward better machining learning. As is known, the LAN ReLU method can largely improve the accuracy of fault diagnosis, but it belongs to strict supervised learning after all, with accuracy severely depending on the available labeled data and the balance among categories. Note that the scarcity of labeled data and the category imbalance are common in real world application, thus the application of the LAN ReLU can be hindered when there are no enough faulty labels. Although the more advanced methods, such as meta-learning [80], transfer learning [81], and multimodal learning [82], [83], can handle such issues through knowledge transfer or multisource integration, they still face various challenges, such as domain similarity and heterogeneous data. Therefore, the present study should be inspiring for resolving such challenges. We hope that the idea of this article can be adopted to enlarge and boost the applicability of more advanced fault diagnosis methods in the near future.

VII. Code Availability

The codes are available in the p-ublic GitHub repository: https://github.com/Dale-Xu/Intelligent-Fault-Classification-Exploration-Inspired-by-SSR.

References

[1] H. Badihi, Y. Zhang, B. Jiang, P. Pillay, and S. Rakheja, "A comprehensive review on signal-based and model-based condition monitoring of wind turbines: Fault diagnosis and lifetime prognosis," *Proc. IEEE*, vol. 110, no. 6, pp. 754–806, Jun. 2022.

[2] M. Chen, R. Qu, and W. Fang, "Case-based reasoning system for fault diagnosis of aero-engines," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117350.

[3] H. Bilal, M. S. Obaidat, M. Shamrooz Aslam, J. Zhang, B. Yin, and K. Mahmood, "Online fault diagnosis of industrial robot using IoRT and hybrid deep learning techniques: An experimental approach," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 31422–31437, Oct. 2024.

[4] R. B. Randall, *Vibration-based Condition Monitoring: Industrial, Automotive and Aerospace Applications*. Hoboken, NJ, USA: Wiley, 2021.

[5] R. Yan and R. X. Gao, "Hilbert–Huang transform-based vibration signal analysis for machine health monitoring," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 6, pp. 2320–2329, Dec. 2006.

[6] V. K. Rai and A. R. Mohanty, "Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform," *Mech. Syst. Signal Process.*, vol. 21, no. 6, pp. 2607–2615, Aug. 2007.

[7] P. W. Tse, Y. H. Peng, and R. Yam, "Wavelet analysis and envelope detection for rolling element bearing fault diagnosis—Their effectiveness and flexibilities," *J. Vib. Acoust.*, vol. 123, no. 3, pp. 303–310, Jul. 2001.

[8] S. N. Chegini, A. Bagheri, and F. Najafi, "Application of a new EWT-based denoising technique in bearing fault diagnosis," *Measurement*, vol. 144, pp. 275–297, Oct. 2019.

[9] J. Altmann and J. Mathew, "Multiple band-pass autoregressive demodulation for rolling-element bearing fault diagnosis," *Mech. Syst. Signal Process.*, vol. 15, no. 5, pp. 963–977, Sep. 2001.

[10] Y. Lei, Z. Qiao, X. Xu, J. Lin, and S. Niu, "An underdamped stochastic resonance method with stable-state matching for incipient fault diagnosis of rolling element bearings," *Mech. Syst. Signal Process.*, vol. 94, pp. 148–164, Sep. 2017.

[11] Y. Zhai, Y. Fu, and Y. Kang, "Incipient bearing fault diagnosis based on the two-state theory for stochastic resonance systems," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[12] Y. Fu, Y. Kang, and R. Liu, "Novel bearing fault diagnosis algorithm based on the method of moments for stochastic resonant systems," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[13] H. Xu, S. Zhou, and T. Yang, "Stochastic resonance of a high-order-degradation bistable system and its application in fault diagnosis with variable speed condition," *Mech. Syst. Signal Process.*, vol. 186, Mar. 2023, Art. no. 109852.

[14] J. Yang, Z. Wang, Y. Guo, T. Gong, and Z. Shan, "A novel noise-aided fault feature extraction using stochastic resonance in a nonlinear system and its application," *IEEE Sensors J.*, vol. 24, no. 7, pp. 11856–11866, Apr. 2024.

[15] Y. Liu, J. Geng, Y. Xu, and H. Zhang, "A novel bearing fault detection by primary resonance of saddle-node bifurcation domains in a hardening duffing oscillator," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.

[16] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, "Stochastic resonance," *Rev. Mod. Phys.*, vol. 70, no. 1, pp. 223–287, 1998.

[17] 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.

[18] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106587.

[19] S. Lu, Q. He, and J. Wang, "A review of stochastic resonance in rotating machine fault detection," *Mech. Syst. Signal Process.*, vol. 116, pp. 230–260, Feb. 2019.

[20] Z. Zhu et al., "A review of the application of deep learning in intelligent fault diagnosis of rotating machinery," *Measurement*, vol. 206, Jan. 2023, Art. no. 112346.

[21] H. Zhao, X. Yang, B. Chen, H. Chen, and W. Deng, "Bearing fault diagnosis using transfer learning and optimized deep belief network," *Meas. Sci. Technol.*, vol. 33, no. 6, Jun. 2022, Art. no. 065009.

[22] 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.

[23] Y. Zhang, T. Zhou, X. Huang, L. Cao, and Q. Zhou, "Fault diagnosis of rotating machinery based on recurrent neural networks," *Measurement*, vol. 171, Feb. 2021, Art. no. 108774.

[24] X. Zhang, C. He, Y. Lu, B. Chen, L. Zhu, and L. Zhang, "Fault diagnosis for small samples based on attention mechanism," *Measurement*, vol. 187, Jan. 2022, Art. no. 110242.

[25] M. Zhao, S. Zhong, X. Fu, B. Tang, and M. Pecht, "Deep residual shrinkage networks for fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4681–4690, Jul. 2020.

[26] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995.

[27] G. An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural Comput.*, vol. 8, no. 3, pp. 643–674, Apr. 1996.

[28] A. Orvieto, H. Kersting, F. Proske, F. Bach, and A. Lucchi, "Anticorrelated noise injection for improved generalization," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 17094–17116.

[29] C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, "Noisy activation functions," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 3059–3068.

[30] X. Liu, L. Duan, F. Duan, F. Chapeau-Blondeau, and D. Abbott, "Enhancing threshold neural network via suprathreshold stochastic resonance for pattern classification," *Phys. Lett. A*, vol. 403, Jul. 2021, Art. no. 127387.

[31] X. Li, "Positive-incentive noise," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 8708–8714, Jun. 2024.

[32] H. Chen, F. Qian, C. Liu, Y. Zhang, H. Su, and S. Zhao, "Training robust deep collaborative filtering models via adversarial noise propagation," *ACM Trans. Inf. Syst.*, vol. 42, no. 1, pp. 1–27, Jan. 2024.

[33] Y. Ren, F. Duan, F. Chapeau-Blondeau, and D. Abbott, "Self-gating stochastic-resonance-based autoencoder for unsupervised learning," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 110, no. 1, Jul. 2024, Art. no. 014107.

[34] K. Audhkhasi, O. Osoba, and B. Kosko, "Noise-enhanced convolutional neural networks," *Neural Netw.*, vol. 78, pp. 15–23, Jun. 2016.

[35] R. Feng, D. Zhao, and Z.-J. Zha, "Understanding noise injection in GANs," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 3284–3293.

[36] Z. Shi, Z. Liao, and H. Tabata, "Enhancing performance of convolutional neural network-based epileptic electroencephalogram diagnosis by asymmetric stochastic resonance," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 9, pp. 4228–4239, Sep. 2023.

[37] S. Huang, H. Zhang, and X. Li, "Enhance vision-language alignment with noise," in *Proc. AAAI Conf. Artif. Intell.*, 2025, vol. 39, no. 16, pp. 17449–17457.

[38] W. Zheng, Q. Zhao, and H. Xie, "Research on adaptive noise mechanism for differential privacy optimization in federated learning," *J. Knowl. Learn. Sci. Technol.*, vol. 3, no. 4, pp. 383–392, Dec. 2024.

[39] Y. Jin, C. Qin, Z. Zhang, J. Tao, and C. Liu, "A multi-scale convolutional neural network for bearing compound fault diagnosis under various noise conditions," *Sci. China Technological Sci.*, vol. 65, no. 11, pp. 2551–2563, Nov. 2022.

[40] C. Yang, Z. Qiao, R. Zhu, X. Xu, Z. Lai, and S. Zhou, "An intelligent fault diagnosis method enhanced by noise injection for machinery," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[41] L. Chen, K. An, D. Huang, X. Wang, M. Xia, and S. Lu, "Noise-boosted convolutional neural network for edge-based motor fault diagnosis with limited samples," *IEEE Trans. Ind. Informat.*, vol. 19, no. 9, pp. 9491–9502, Sep. 2023.

[42] L. Xiao, J. Wang, X. Liu, H. Sun, and H. Zhao, "A novel fault diagnosis method based on convolutional neural network with adaptive noise injection," *Meas. Sci. Technol.*, vol. 36, no. 3, Mar. 2025, Art. no. 036101.

[43] Z. Xu, Y. Fu, R. Mei, Y. Zhai, and Y. Kang, "Novel classification algorithms inspired by firing rate stochastic resonance," *Nonlinear Dyn.*, vol. 113, no. 1, pp. 497–517, Jan. 2025.

[44] J. Suo, H. Wang, X. Shen, Y. Yan, and H. Dong, "Mutual information-assisted feed-forward cascaded stochastic resonance for large parameter," *Nonlinear Dyn.*, vol. 111, no. 20, pp. 19225–19247, Oct. 2023.

[45] Z. Xu, Y. Zhai, and Y. Kang, "Mutual information measure of visual perception based on noisy spiking neural networks," *Frontiers Neurosci.*, vol. 17, Aug. 2023, Art. no. 1155362.

[46] S. Durrant, Y. Kang, N. Stocks, and J. Feng, "Suprathreshold stochastic resonance in neural processing tuned by correlation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, no. 1, Jul. 2011, Art. no. 011923.

[47] D. Rousseau, F. Duan, and F. Chapeau-Blondeau, "Suprathreshold stochastic resonance and noise-enhanced Fisher information in arrays of threshold devices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 3, Sep. 2003, Art. no. 031107.

[48] N. G. Stocks, "Suprathreshold stochastic resonance in multilevel threshold systems," *Phys. Rev. Lett.*, vol. 84, no. 11, pp. 2310–2313, Mar. 2000.

[49] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[50] A. Maas, A. Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 16–21.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[52] F. Chapeau-Blondeau, S. Blanchard, and D. Rousseau, "Noise-enhanced Fisher information in parallel arrays of sensors with saturation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 031102.

[53] T. M. Cover and J. A. Thomas, "Information theory and the stock market," in *Elements of Information Theory*. New York, NY, USA: Wiley, 1991, pp. 543–556.

[54] R. B. Randall, J. Antoni, and S. Chobsaard, "The relationship between spectral correlation and envelope analysis in the diagnostics of bearing faults and other cyclostationary machine signals," *Mech. Syst. Signal Process.*, vol. 15, no. 5, pp. 945–962, Sep. 2001.

[55] R. B. Randall and J. Antoni, "Rolling element bearing diagnostics—A tutorial," *Mech. Syst. Signal Process.*, vol. 25, no. 2, pp. 485–520, 2010.

[56] 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.

[57] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[58] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[59] Z. Feng, M. Liang, and F. Chu, "Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples," *Mech. Syst. Signal Process.*, vol. 38, no. 1, pp. 165–205, Jul. 2013.

[60] X. Zhang, H. Wang, C. Wang, M. Liu, and G. Xu, "Time-segment-wise feature fusion transformer for multi-modal fault diagnosis," *Eng. Appl. Artif. Intell.*, vol. 138, Dec. 2024, Art. no. 109358.

[61] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mech. Syst. Signal Process.*, vol. 100, pp. 439–453, Feb. 2018.

[62] L. Zou, X. Zhu, C. Wu, Y. Liu, and L. Qu, "Spectral–spatial exploration for hyperspectral image classification via the fusion of fully convolutional networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 659–674, 2020.

[63] S. Sengupta et al., "A review of deep learning with special emphasis on architectures, applications and recent trends," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105596.

[64] M. M. Taye, "Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions," *Computers*, vol. 12, no. 5, p. 91, Apr. 2023.

[65] S. Ayas and M. S. Ayas, "A novel bearing fault diagnosis method using deep residual learning network," *Multimedia Tools Appl.*, vol. 81, no. 16, pp. 22407–22423, Jul. 2022.

[66] Z. Dong, D. Zhao, and L. Cui, "Rotating machinery fault classification based on one-dimensional residual network with attention mechanism and bidirectional gated recurrent unit," *Meas. Sci. Technol.*, vol. 35, no. 8, Aug. 2024, Art. no. 086001.

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[68] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong, and J. Hu, "Limited data rolling bearing fault diagnosis with few-shot learning," *IEEE Access*, vol. 7, pp. 110895–110904, 2019.

[69] Y. Dong, Y. Li, H. Zheng, R. Wang, and M. Xu, "A new dynamic model and transfer learning based intelligent fault diagnosis framework for rolling element bearings race faults: Solving the small sample problem," *ISA Trans.*, vol. 121, pp. 327–348, Feb. 2022.

[70] C. Zhao, E. Zio, and W. Shen, "Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study," *Rel. Eng. Syst. Saf.*, vol. 245, May 2024, Art. no. 109964.

[71] G. Fu, X. Wang, Y. Liu, and Y. Yang, "A robust bearing fault diagnosis method based on ensemble learning with adaptive weight selection," *Expert Syst. Appl.*, vol. 269, Apr. 2025, Art. no. 126420.

[72] A. Vaswani et al., "Attention is all you need," in *Proc. NIPS*, 2017, pp. 6000–6010.

[73] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[74] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[75] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[76] B. J. Kim, H. Choi, H. Jang, D. Lee, and S. W. Kim, "How to use dropout correctly on residual networks with batch normalization," in *Proc. Uncertainty Artif. Intell.*, 2023, pp. 1058–1067.

[77] M. Fortunato et al., "Noisy networks for exploration," 2017, *arXiv:1706.10295*.

[78] X. Chen, B. Zhang, and D. Gao, "Bearing fault diagnosis base on multi-scale CNN and LSTM model," *J. Intell. Manuf.*, vol. 32, no. 4, pp. 971–987, Apr. 2021.

[79] Y. Kang, Y. Fu, and Y. Chen, "Signal-to-noise ratio gain of an adaptive neuron model with gamma renewal synaptic input," *Acta Mechanica Sinica*, vol. 38, no. 1, Jan. 2022, Art. no. 521347.

[80] C. Li, S. Li, H. Wang, F. Gu, and A. D. Ball, "Attention-based deep meta-transfer learning for few-shot fine-grained fault diagnosis," *Knowledge-Based Syst.*, vol. 264, Mar. 2023, Art. no. 110345.

[81] I. Misbah, C. K. M. Lee, and K. L. Keung, "Fault diagnosis in rotating machines based on transfer learning: Literature review," *Knowl.-Based Syst.*, vol. 283, Jan. 2024, Art. no. 111158.

[82] L. Cheng, Z. An, Y. Guo, M. Ren, Z. Yang, and S. McLoone, "MMFSL: A novel multimodal few-shot learning framework for fault diagnosis of industrial bearings," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[83] Y. Wang, Y. Lei, N. Li, X. Li, and B. Yang, "Multimodal correlation-aware fusion framework for enhanced machinery health prognosis with unlabeled and low-quality data exploitation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 7, pp. 12040–12051, Jul. 2025.