Hindawi
Shock and Vibration
Volume 2024, Article ID 3374107, 16 pages
https://doi.org/10.1155/2024/3374107

Research Article

Deep Multiscale Soft-Threshold Support Vector Data Description for Enhanced Heavy-Duty Gas Turbine Generator Sets' Anomaly Detection

Zhang Kun,[1,2] **Li Hongren**,[1] **Wang Xin**,[1] **Xie Daxing**,[1] **and Sun Xiaokai**[1]

[1] *Huadian Electric Power Research Institute Co., Ltd., Hangzhou, China*
[2] *College of Electrical Engineering, Zhejiang University, Hangzhou, China*

Correspondence should be addressed to Zhang Kun; 3240569595@qq.com

Received 28 December 2023; Revised 4 March 2024; Accepted 15 April 2024; Published 29 April 2024

Academic Editor: Antonio Giuffrida

This paper introduces an innovative approach, Deep Multiscale Soft-Threshold Support Vector Data Description (DMS-SVDD), designed for the detection of anomalies and prediction of faults in heavy-duty gas turbine generator sets (GENSETs). The model combines a support vector data description (SVDD) with a deep autoencoder backbone network framework, integrating a multiscale convolutional neural network (M) and soft-threshold activation network (S) into the Deep-SVDD framework. In comparison with conventional methods, such as One-Class Support Vector Machine (OCSVM) and autoencoder (AE), DMS-SVDD demonstrates improvements in accuracy (by 22.94%), recall (by 32%), F1 score (by 12.02%), and smoothness (by 39.15%). The model excels particularly in feature extraction, denoising, and early fault detection, offering a proactive strategy for maintenance. Furthermore, the DMS-SVDD demonstrated enhanced training efficiency with a reduction in the convergence rounds by 66% and overall training times by 34.13%. The study concludes that DMS-SVDD presents a robust and efficient solution for gas turbine anomaly detection, with practical advantages for decision support in turbine maintenance. Future research could explore additional refinements and applications of the DMS-SVDD model across diverse industrial contexts.

1. Introduction

Power generators, particularly thermal power generators, play a significant role in the global power market. In regions such as China, heavy-duty gas turbine generator sets (GENSETs) are commonly utilized as peak load units, beyond their original continuous service designation. The frequent start-stop cycles of these GENSETs, due to the variable load, can lead to a reduction in their lifespan. Unforeseen malfunctions, anomalies, degradation, and faults during operation can further impact the reliability and safety of GENSETs. As a result, anomaly diagnosis is essential for maintaining the optimal working status of GENSETs.

The increasing complexity of engineering systems, along with the emphasis on safety and cost-effectiveness, has underscored the need for reliable, efficient, and autonomous diagnostic and health monitoring systems. These systems should be capable of real-time interaction with human experts, surpassing traditional statistical trend analysis and out-of-species monitoring techniques. With advancements in data acquisition capabilities, there is a growing shift toward data-driven approaches for anomaly diagnosis. These methods, based on statistical learning, regression, and neural networks, offer simpler forms and require less project work, making them increasingly popular in both academia and industry [1–4].

Traditional algorithms such as support vector machines, linear regression, and neural networks play a significant role in the data-based anomaly diagnosis of dynamic systems. However, the implementation of supervised learning algorithms in fault diagnosis presents challenges, particularly in industrial settings where data are often unlabeled and abnormal data are scarce and hidden within large volumes of normal data [5, 6]. The complex configuration of gas turbines, along with the interference experienced by its components during monitoring, results in a diverse array of signal data with intertwined internal fault modes and significant redundant noise, complicating the extraction of fault characteristics [7–15].

A method for assessing rolling-bearing degradation is proposed to address the issues of limited information in single vibration features and inaccurate high-dimensional feature sets. The approach involves adaptive sensitive feature selection and a multistrategy optimized SVDD [16]. Fan et al. [17] developed an unsupervised anomaly detection method for intermittent time-series in manufacturing enterprises. It utilizes a new abnormal fluctuation similarity matrix and agglomerative hierarchical clustering to identify the anomalies. The support vector data description model was used for feature extraction and hypersphere training, enabling the detection of abnormal points at a micro-granularity level. Navarro-Acosta et al. [18] applied a fault detection system that combines SVDD with metaheuristic algorithms to a real-world industrial process with a limited number of measured faults. The primary contribution of this research is the comparison of various swarm intelligence algorithms for effectively optimizing the SVDD hyperparameters. The ensemble deep-SVDD (EDeSVDD) method was introduced in Ref. [19] for improved anomaly detection and more effective monitoring of process faults. It utilizes a DeSVDD framework with a multilayer feature extraction structure and regularization of deep network weights. Bayesian inference-based ensemble learning was employed to generate DeSVDD submodels at the parameter and structure levels, integrating them for comprehensive monitoring. Fan et al. [20] proposed a novel hybrid method for key performance indicator anomaly detection based on the VAE and SVDD. BiLSTM and batch normalization are introduced into the VAE reconstruction module to capture time correlation and prevent divergence. The method utilizes EWMA to smooth reconstruction errors and reduce false positives and false negatives. In the SVDD anomaly detection module, smoothed reconstruction errors were used to train the SVDD and adaptively determine the anomaly detection threshold.

However, the traditional SVDD encounters the challenge of lacking labeled information for anomalous samples during the training phase. This deficiency complicates the distinction between the normal and anomalous patterns throughout the learning process, especially when the distribution of anomalous patterns closely aligns with that of the normal patterns. Unlabeled data typically introduce noise, and SVDD demonstrates increased sensitivity to such noise. In the absence of labeled information, determining which data genuinely represent anomalies, as opposed to mere noise, becomes exceedingly challenging, thus impacting the precision of the model in anomaly detection. This paper introduces a construction approach for an advanced fault warning model, the Deep Multiscale Soft-Threshold Support Vector Data Description (DMS-SVDD). The model was designed based on a deep AE backbone network architecture that integrates information from diverse sources [10, 21, 22]. To address the challenge of extracting fault features from unlabeled data, the original autoencoder network is enhanced with a multiscale convolutional feature extraction module, refining features and improving the model's capability to represent the intricate samples. To mitigate the issue of noise redundancy, a soft-threshold activation module was integrated into the network to eliminate noise, thereby enhancing the accuracy and stability of the model in recognizing patterns. Experimental validation using real operational data from a specific gas turbine illustrates the superiority of the proposed enhanced model in terms of monitoring the accuracy, noise reduction, and training efficiency when compared to the original model. A comparative analysis with classical unsupervised learning methods, such as the OCSVM and AE, further corroborates the effectiveness of the proposed approach in augmenting the anomaly detection capabilities of GENSETs. The following sections of this article provide a comprehensive exploration of the proposed anomaly detection approach for GENSETs. In Section 2, related theories are discussed, with a focus on the SVDD model and the Deep Autoencoder Backbone Network Framework. Furthermore, the chapter outlines the incorporation of these theories into the proposed the Deep-SVDD Network Model, offering a thorough comprehension of the model's architecture and training procedures. In Section 4, a case study is presented that employs data from GENSET, with a specific emphasis on preprocessing sensor data from various components. The case study includes temporal waveform representations of diverse parameters, such as engine speed and temperatures, accompanied by a detailed description of the process, dataset division, and normalization procedures. In addition, this section discusses the experimental platform, hyperparameter configuration, and evaluation metrics used to assess the performance of the proposed anomaly detection model. Section 4 presents the results and discussions of the ablation experiments and compares the proposed DMS-SVDD method with other anomaly detection algorithms. The evaluation metrics, confusion matrices, and efficiency analysis highlight the superior performance and efficiency of the DMS-SVDD model, demonstrating its effectiveness in anomaly detection and training efficiency compared to alternative methods. Finally, Section 5 summarizes the findings of the investigation, highlighting superior performance of the DMS-SVDD approach in gas turbine anomaly detection.

2. Related Theories

2.1. Support Vector Data Description Model. SVDD is a single-value classification algorithm based on statistical theory and the principle of structural risk minimization. It aims to distinguish between the target and nontarget samples and is commonly applied in fields such as anomaly detection and intrusion detection [23–25]. Because data are typically linearly inseparable under normal circumstances, SVDD first performs nonlinear mapping on the original training samples X, projecting them into a high-dimensional feature space. Subsequently, in this feature space, SVDD seeks to determine the minimum-volume hypersphere, which is referred to as the optimal hypersphere. This hypersphere encompasses all or most of the training samples mapped to the feature space. The optimal hypersphere can be described by its center and radius, providing a representation of the boundary of the normal samples. The basic concept is illustrated in Figure 1.

Based on the considerations mentioned above, we assume that we have samples N, denoted by $x_1, x_2, \ldots, x_n$. We can formulate the following constrained optimization problem:

$$\begin{aligned} &\min\left(R^2 + c\sum_{i=1}^{m}\xi_i\right), \\ &s.t.\ \|\phi(x_i) - a\|^2 \le R^2 + \xi_i, \xi_i \ge 0, \quad \forall i = 1, 2, \cdots n, \end{aligned} \tag{1}$$

where a, R, and ξ_i indicate the hypersphere center, radius, and relaxation factor, respectively, and each training sample x_i should satisfy the condition of being inside or on the boundary of the hypersphere. $\phi(x_i)$ represents the nonlinear mapping of the training sample x_i to the high-dimensional feature space. This optimization problem aims to achieve a compact and representative description of the data distribution in the feature space.

In this constrained optimization problem, the objective function comprises two terms: the first term seeks to minimize the radius of the hypersphere (representing structural risk), and the second term aims to provide a hypersphere with a certain degree of tolerance (representing empirical risk). Ideally, when all data points lie within a hypersphere, the empirical risk becomes zero. Therefore, we introduce the following constraints to this optimization problem $\|\phi(x_i) - a\|^2 \le R^2 + \xi_i$ and $\xi_i \ge 0$. The introduction of a relaxation factor enhances the robustness of the model to mitigate the impact of individual extreme data points. Parameter c acts as a penalty factor that strikes a balance between the volume of the hypersphere and misclassification rate, thus enabling control over the magnitude of the influence exerted by the relaxation factor.

By solving (1), the center and radius of the hypersphere are obtained. For a new data point x', its distance d from the center of the hypersphere is calculated using the following formula:

$$d = \sqrt{\|x' - a\|^2}. \tag{2}$$

If $d \ge R$, the data point is located outside the hypersphere, indicating that it can be classified as an abnormal value. If $d \le R$, the data point is located inside the hypersphere, indicating that it can be classified as a normal value.

2.2. Deep Autoencoder Backbone Network Framework.

2.2. Deep Autoencoder Backbone Network Framework. An autoencoder is an unsupervised method for data dimensionality reduction and feature representation [26–29]. A deep autoencoder network consists of an encoder function $f_\theta$and decoder function g_θ. For a dataset $\{x_i x_1, x_2, \cdots x_n\}$, the encoding process learns the hidden features of the input signals $h = f_\theta(x)$, where h is the feature vector of the input data x, achieving dimensionality reduction and feature extraction in the hidden layers. The decoding process maps the features back into the input space, thereby producing a reconstructed feature vector $z = g_\theta(x)$. The output signal has the same dimensions as the input signal. Autoencoders obtain robust feature representations by comparing the differences between inputs and outputs. They can extract features from signals and reduce the dimensionality. During the training process of the autoencoder backbone network, the encoder and decoder were jointly trained to find the parameter vectors that minimized the reconstruction error. Effective encoding of the original data were obtained by learning in the hidden layers

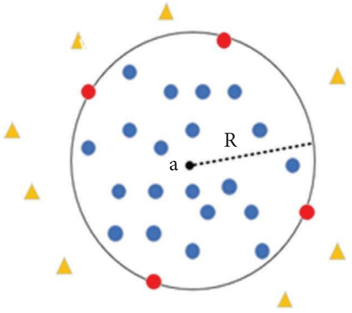


Figure 1: The basic idea of SVDD.

$$\theta = \operatorname{argmin}\sum_{i=1}^{n} L(x_i, z_i), \tag{3}$$

where θ represents the model parameters, a set of values that we aim to find through the optimization process to minimize the total sum of the loss function L. In other words, θ encompasses all the weights and bias terms that need to be learned within the model. In deep learning, θ is typically updated iteratively through optimization algorithms such as backpropagation and gradient descent, with the goal of finding a set of parameters that allow the model to make predictions on the training data as accurately as possible. In the context of autoencoders, θ can include the parameters of both the encoder and decoder. $L(x_i, z_i)$ denotes the reconstruction error function, typically calculated using either the mean squared error (MSE) function or the cross-entropy function [30].

2.3. Deep Support Vector Data Description Network Model. Traditional support vector data description (SVDD) methods have been widely used for anomaly detection due to their effectiveness in capturing the boundary of normal data in a feature space. However, these methods have certain limitations that can impact their performance and applicability in various scenarios. One of the main drawbacks of traditional SVDD is its poor scalability. As the size of the dataset increases, the computational complexity of SVDD increases significantly, making it challenging to apply to large-scale data. Additionally, traditional SVDD methods are limited by their dimensional constraints. They often struggle to handle high-dimensional data effectively, as the complexity of the boundary enclosing the normal data increases with the dimensionality of the feature space. This can lead to suboptimal performance in scenarios where the data have a large number of features. Furthermore, traditional SVDD methods rely heavily on the choice of the kernel function, which can be difficult to tune for different types of data, leading to further limitations in their flexibility and adaptability. To overcome the drawbacks of traditional SVDD methods, such as poor scalability and dimensional limitations, this study proposes a new approach for anomaly detection based on neural networks and Deep-SVDD. The core concept is to use neural networks to extract kernel features for SVDD, leveraging the learned center of the normal features from normal data to identify anomalies. The neural network maps normal data into a hypersphere with the minimum volume defined by center a and radius R (the objective is to determine a and R). The mapping of normal data resides within the surface of the hypersphere, whereas the mapping of anomalous data lies outside the surface of the hypersphere.

The left-hand side of Figure 2 illustrates the original data points. For simplicity, in the analysis, it is assumed that the feature vectors of the data are two-dimensional, representing an input space. The objective was to enclose normal data samples using a hypersphere with the smallest possible radius. The Deep-SVDD approach involves training a neural network to transform the input data into the output space depicted on the right-hand side. When a new data point is processed through the model and falls within a circle (known as a hypersphere in high-dimensional space) with center c and radius R, it is classified as normal (blue circles). Conversely, if it falls outside of the circle, it is classified as an anomaly (yellow triangles). In mapping all normal samples onto the hypersphere, the red circle represents the normal sample point furthest from the center of the hypersphere, thereby determining the radius of the hypersphere. Additionally, the distance between the data point and the center of the circle denotes the severity of the anomaly, with greater distances indicating more severe anomalies.

Moreover, when constructing a Deep-SVDD network directly using a neural network, there are certain requirements for the initial parameters during training. First, the initial vector of center a must not be set to 0. Second, each neuron in the neural network must not have a bias term, b. Finally, bounded activation functions should not be used. Failure to meet these conditions may cause Deep-SVDD to map all data samples to the same point during training, minimizing the volume of the hypersphere and resulting in a collapsed hypersphere with a radius R of 0. By combining an autoencoder with the Deep-SVDD network, the neural network maps the data points closer to the center while reconstructing them to resemble the original signals as much as possible. If the encoder maps all data points to a single point, the decoder cannot reconstruct distinct input signals, thereby effectively preventing the collapse of the hypersphere. The network structure depicted in Figure 3 was obtained by integrating the autoencoder with the SVDD model.

An autoencoder consists of an encoder and a decoder. The encoder projects the original data into a low-dimensional feature space to perform the feature extraction and dimensionality reduction. By contrast, the decoder attempts to reconstruct the original data from the projected low-dimensional space. The parameters of both networks are learned using a reconstruction loss function. The reconstruction cost function is defined as follows:

$$L = \sum_{i=1}^{n} \|x_i - x_i'\|. \tag{4}$$

The hidden layers of the autoencoder perform dimensionality reduction on the input, thereby guiding the neural network to extract features from the dataset. The autoencoder network is trained using a normal dataset, denoted as $X = \{x_i\}_{i=1}^{n}$, to obtain the hidden layer features $Z = \{z_i\}_{i=1}^{n}$. The dimensionality of each z_i in the hidden layer can be designed based on specific requirements or characteristics of the data. x represents the original sample points, while x' denotes the reconstructed sample points obtained after passing through the encoder and decoder. The encoder is extracted separately and optimized using a specific cost function. The cost function is defined as follows:

$$L = \frac{1}{n}\sum_{i=1}^{n} \|z_i - a\|^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|W^l\|^2, \tag{5}$$

where a is the center of the hypersphere, and $a = 1/\mathrm{n}\sum_{i=1}^{n} z^i$. The first term in the equation applies quadratic loss to penalize the distance of each data point to the center of the hypersphere, with the aim of minimizing the volume of the hypersphere. The second term represents the weight decay regularization in the network, which helps to prevent overfitting. λ is a hyperparameter that controls the strength of the regularization term. Deep-SVDD aims to shrink the hypersphere by minimizing the average distance of all data representations to the center. Through iterative training of the network, we eventually obtained the feature space mapping, as well as the center a and radius R of the hypersphere. Create a scoring function that measures the anomaly score

$$d = \|z_i - a\|^2. \tag{6}$$

The distance between the new data points and the center of the hypersphere obtained through the network model was utilized to determine whether the data were anomalous. Additionally, the distance can provide insights into the degree of abnormality in data [31–36].

2.4. Model Multiscale Convolutional Neural Networks. As the signals are sourced from different components and systems, they lack a consistent scale. To effectively capture multiscale information from diverse sources at both the macro- and microlevels, the entire model was designed as a multiscale hierarchical structure. The encoding section of the model architecture includes the parallel concatenations of convolutional kernels of varying sizes.

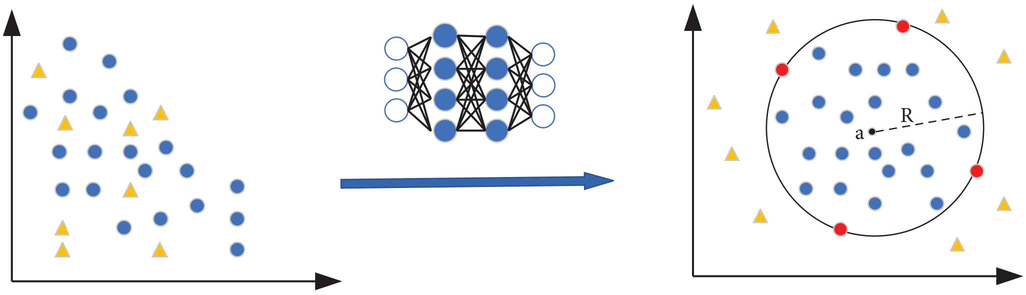


Figure 2: The basic idea of Deep-SVDD.

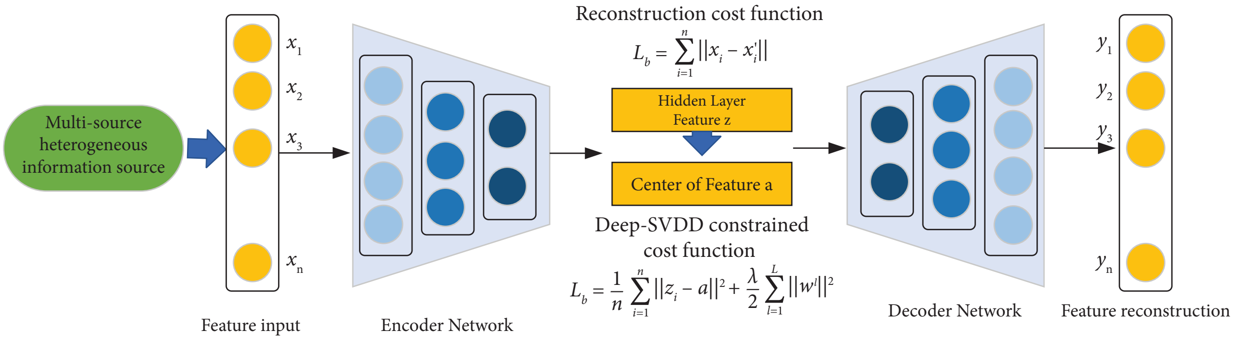

Multi-source heterogeneous information source
x_1
x_2
x_3
x_n
Feature input
Encoder Network
Reconstruction cost function
$L_b = \sum_{i=1}^{n} ||x_i - x_i^{'}||$
Hidden Layer Feature z
Center of Feature a
Deep-SVDD constrained cost function
$L_b = \frac{1}{n}\sum_{i=1}^{n} ||z_i - a||^2 + \frac{\lambda}{2}\sum_{l=1}^{L} ||w^l||^2$
Decoder Network
y_1
y_2
y_3
y_n
Feature reconstruction

Figure 3: Deep-SVDD network framework.

Convolutional neural networks (CNNs) are particularly well-suited for feature extraction in the encoding section of an autoencoder due to their ability to preserve spatial relationships and capture local patterns in the input data. By employing a multiscale CNN architecture, we can efficiently extract signal features at various scales, thereby enhancing the model's expressive capacity and generative effectiveness. Utilizing the feature extraction capabilities of these distinct kernels progressively captures multiscale information. Subsequently, the multiscale features acquired by the parallel convolutional units are combined to create the input data for the subsequent decoding section. In a multilayered network structure, a multiscale CNN can efficiently extract signal features at various scales, thereby enhancing the model's expressive capacity and generative effectiveness.

2.5. Soft-Threshold Activation Network. Soft thresholding is a widely used technique for feature filtering in various fields such as signal processing, statistics, and machine learning [37–41]. It involves setting features with values close to 0 directly by establishing a feature threshold, while retaining crucial features exceeding the threshold. This method was designed to effectively suppress the noise. The soft thresholding function is represented by the following equation:

$$S_{Th} = \operatorname{sgn}(s) \cdot \max[(|s| - Th), 0], \tag{7}$$

where s represents the input signal, s_{Th} denotes the soft-threshold signal, sgn(·) is the sign function, and Th is the threshold [42–44].

Under the influence of the soft thresholding function, when the threshold is set to $Th = \tau$, the input signal is zeroed within the threshold range $|s| < \tau$ and shifted outside the threshold range, thereby reducing the impact of signal values near zero. This enhances the "contrast" of internal data in the signal, akin to elevating the importance of critical data points, allowing the network to focus on clean data. As illustrated in Figure 4, the gradient of the function is 1 when the absolute value of the input signal exceeds the threshold, and 0 otherwise. This behavior is similar to the ReLU activation function depicted in Figure 5 and effectively prevents the vanishing gradient problem, facilitating gradient backpropagation. Furthermore, compared with the ReLU function, it better preserves the negative features of the input signal, thus preventing information loss.

For soft thresholding operations, the selection of the threshold significantly influences the denoising effectiveness of the signal. Therefore, the key is to determine an appropriate threshold. However, manually setting the threshold is not only time-consuming but also challenging to guarantee the final outcome. Neural network models, with the advantage of data-driven intelligent learning, circumvent manual intervention and can serve as an effective means for threshold learning. To address the issue of severe noise in gas turbine monitoring signals, which complicates pattern recognition during anomaly detection, this study integrates attention mechanisms with the soft thresholding model. This established a soft-threshold activation network, thereby mitigating the interference of redundant noise in the signal.

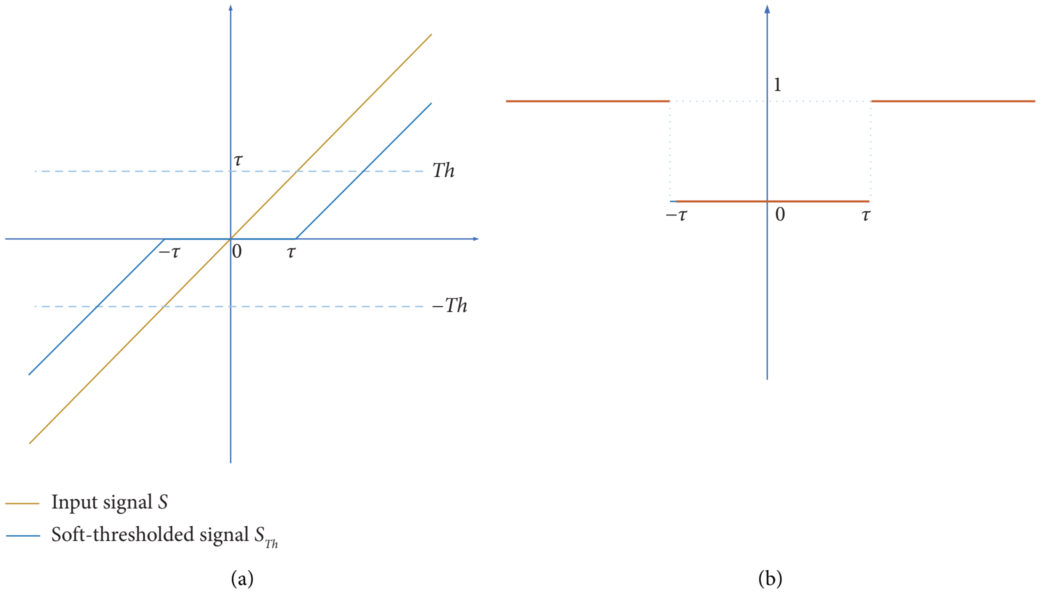


Figure 4: Soft-threshold activation function: (a) soft-threshold function curve and (b) soft-threshold gradient.

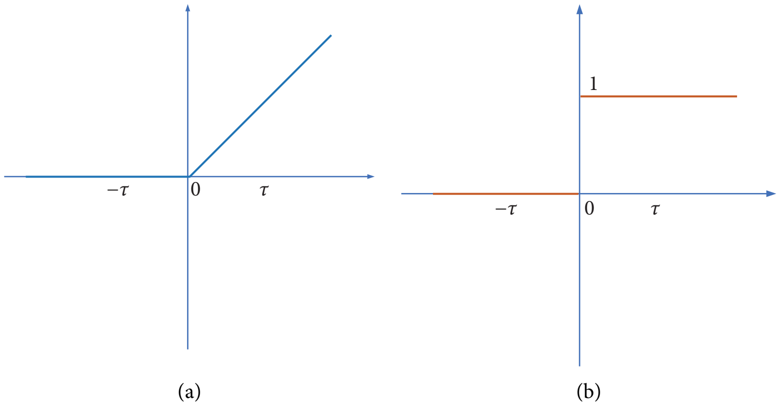


Figure 5: ReLU activation function: (a) ReLU function curve and (b) ReLU activation function gradient.

Figure 6 depicts the designed soft-threshold activation network, where ⊗ represents the element-wise multiplication of the matrices. In this module, the absolute values of all features in the absolute value layer are first input, and the absolute values of the obtained input undergo global average pooling to obtain a feature, denoted as A. In another path, the feature output of the global average pooling layer is fed into a subnetwork with the sigmoid function as its last layer. Consequently, the network' output was normalized to the range of 0 to 1, and the output of the fully connected layer was denoted as α. The final threshold can be expressed as $\alpha \times A$. Therefore, the threshold is a number between 0 and 1, multiplied by the average value of the absolute values of the feature map. This ensures that the threshold is positive and not excessively large and allows the model to continuously adjust the local weights of each feature. This enhancement strengthens the ability of the model to express features for the current task, thereby eliminating the impact of noise on the recognition model.

2.6. DMS-SVDD Network. Based on the Deep-SVDD model, the DMS-SVDD model was constructed by employing a multiscale convolutional neural network (CNN) to replace the encoding module of the deep autoencoder for feature extraction. The rationale behind this modification is twofold:

Enhanced Feature Extraction: Multiscale CNNs are capable of capturing features at different spatial scales, which is particularly beneficial for processing time-series data like vibration signals. By using convolutional kernels of varying sizes (e.g., 1×3, 1×5, and 1×7), the model can extract a richer set of features, encompassing both local and global patterns. This is crucial for accurately identifying anomalies in complex systems.

Improved Noise Reduction: The integration of a soft-threshold activation network within the DMS-SVDD model further enhances its ability to eliminate noise and redundant information. By selectively attenuating features that are irrelevant to the current task, the model can focus on the

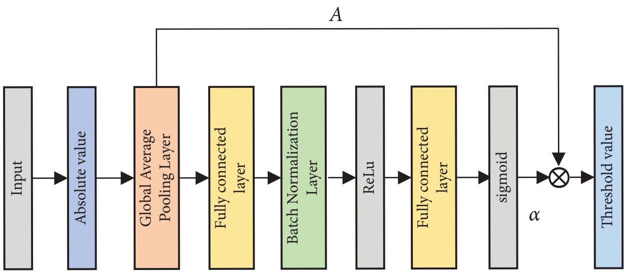


Figure 6: Soft-threshold activation network process diagram.

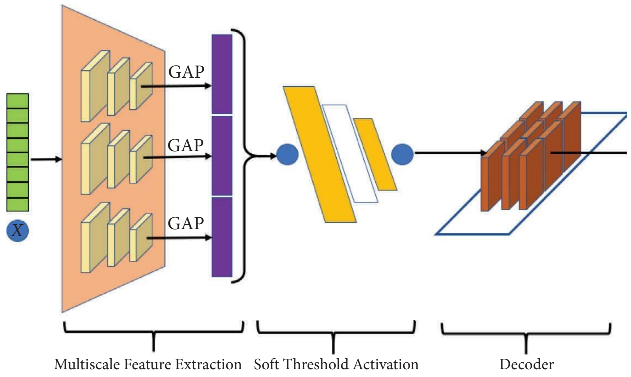


Figure 7: DMS-SVDD network framework.

most pertinent features, leading to more accurate and reliable anomaly detection. The model is divided into three components, as illustrated in Figure 7.

(a) Multiscale feature extraction. The original vibration data are directly fed into three parallel branches of convolutional neural networks with different-sized kernels: 1 × 3, 1 × 5, and 1 × 7. Varying kernel sizes allow for the extraction of features at different scales, and multiple layers of convolutional networks are employed to capture features at different levels. The features extracted at different scales are then subjected to global average pooling, channelwise, and concatenated along the channels to achieve the integration of deep-level features.

(b) Soft-threshold activation. Fused features are input into a soft-threshold activation network that learns distinct channelwise threshold vectors. These vectors are then applied to different channels, employing the soft-threshold function to selectively attenuate features irrelevant to the current task by setting them to 0. Simultaneously, the features relevant to the task are preserved and outputted. During the training process, the model dynamically adjusts the thresholds to minimize the difference between the model output and ground truth.

(c) Decoder. By utilizing the encoded features from the hidden layer, the decoder reconstructs the original sample data through a mapping process. Subsequently, the performance of the model is assessed by evaluating the similarity between the reconstructed and original samples to gauge the reconstruction capability of the autoencoder model.

The parameter configuration for the encoder backbone network of the DMS-SVDD model is outlined in Table 1. Given that the inputs consist exclusively of one-dimensional time-series data, the network employs one-dimensional convolutional kernels. The notation Cov1d (133, 128, 3, 2, 1) signifies the use of one-dimensional convolutional kernels, with an input channel count of 133, an output channel count of 128, a kernel size of 1 × 3, a stride of 2, and a padding of 1. To capture the multiscale temporal features of the monitoring data, three parallel submodules of convolutional neural networks with distinct kernel sizes were designed. After the convolutional layer block, the length of the time-series progressively diminishes layer-by-layer, while concurrently, the channel count undergoes continuous variation. Subsequently, global average pooling was applied channelwise to capture the data features for each channel. In the final stage of the multiscale feature extraction module, a concatenation layer was employed to integrate the features extracted by the convolutional kernels of different sizes. Subsequently, the extracted features are input into a soft-threshold module to learn distinct thresholds. Ultimately, the input feature map undergoes decoding processing through an upsampling operation. The upsampling operation is employed to increase the spatial resolution of the encoded feature map, effectively reversing the dimensionality reduction performed during the encoding process. This is achieved by interpolating additional points between the existing data points in the feature map, thereby expanding its size. In the context of our DMS-SVDD model, the upsampling operation is applied to the encoded features to reconstruct the original sample data with the same dimensions as the input signal.

The decoder employs a series of upsampling layers, each followed by a convolutional layer. The upsampling layers increase the temporal resolution of the feature map, while the convolutional layers refine the upsampled features to ensure that the reconstructed signal closely resembles the original input. The combination of upsampling and convolutional layers in the decoder allows for a gradual and controlled reconstruction of the input signal from the compressed feature representation.

By utilizing this decoding process, the model is able to assess the reconstruction capability of the autoencoder by evaluating the similarity between the reconstructed and original samples. A high degree of similarity indicates that the model has effectively captured the essential features of the input data, while a low similarity suggests that the model may have failed to accurately encode and decode the input signal.

The DMS-SVDD model underwent training solely with normal sample data. The network training process comprises two distinct phases: the autoencoder network training phase and feature hypersphere construction phase. In the initial phase, feature extraction and dimensionality reduction were accomplished through an autoencoding structure. In the subsequent phase, the features extracted by the encoder are

Table 1: Configuration of the main encoder backbone network parameters.

Layer	Algorithm structure	Parameters
Branch 1	Convolution, batch normalization, ReLU activation	Cov1d (133, 128, 3, 2, 1)
	Convolution, batch normalization, ReLU activation	Cov1d (128, 64, 3, 2, 1)
	Convolution, batch normalization, ReLU activation	Cov1d (64, 32, 3, 2, 1)
Branch 2	Convolution, batch normalization, ReLU activation	Cov1d (133, 128, 5, 2, 2)
	Convolution, batch normalization, ReLU activation	Cov1d (128, 64, 5, 2, 2)
	Convolution, batch normalization, ReLU activation	Cov1d (64, 32, 5, 2, 2)
Branch 3	Convolution, batch normalization, ReLU activation	Cov1d (133, 128, 7, 2, 3)
	Convolution, batch normalization, ReLU activation	Cov1d (128, 64, 7, 2, 3)
	Convolution, batch normalization, ReLU activation	Cov1d (64, 32, 7, 2, 3)
GAP layer	Global average pooling	C:32
Feature fusion layer	Feature concatenation	C:96
Soft thresholding activation layer	Soft thresholding activation module	C:96

exclusively utilized to construct a hypersphere representing a healthy state, with a focus on minimizing its volume. Following the completion of training, anomaly determination for new samples can be performed based on the distance between their features and the center of the hypersphere.

2.7. Model Training. The training of models is segmented into two distinct phases, where the loss incurred during model training is analogous to that described by a network model employing deep support vector data.

In the initial phase, the backbone of the AE network is trained with the optimization objective of preserving the similarity between the original input data and the reconstructed data. This is achieved through a reconstruction loss function designed to learn and optimize the model, thereby extracting the corresponding latent features. The reconstruction loss function is defined as follows:

$$L = \sum_{i=1}^{n} \|x_i - \widehat{x}_i\|. \tag{8}$$

The hidden layer representation of AE is a dimensionality reduction of the input, which, in turn, guides the neural network in extracting features from the dataset. By training the AE network with a normal dataset $X = \{x_i\}_{i=1}^{n}$, we obtain the hidden layer features $Z = \{z_i\}_{i=1}^{n}$, where $n = 96$, indicating that the hidden layer features are 96-dimensional.

The algorithm for the first phase is presented in Table 2, in a pseudocode format resembling Algorithm 1.

In the second phase, the trained encoder is utilized to extract hidden features and perform hypersphere optimization. The optimization objective is to adjust the model parameters such that normal samples cluster near the center of the sphere, while anomalous samples are pushed away from the center, thereby minimizing the volume of the hypersphere. The loss function is as follows:

$$L = \frac{1}{n}\sum_{i=1}^{n} \|z_i - a_i\|^2 + \frac{\lambda}{2}\sum_{l=1}^{L} \|w^l\|^2, \tag{9}$$

where $a = 1/\mathrm{n}\sum_{i=1}^{n} z^i$ represents the center of the sphere. The first term in the formula employs a quadratic loss to penalize the distance of each data point from the center of the hypersphere, thus minimizing the volume of the hypersphere. The second term, where W represents the parameters of the network model, acts as a network weight decay regularization term to prevent overfitting, with λ being a hyperparameter. Deep-SVDD minimizes the average distance of all data representations to the center, effectively shrinking the sphere. Through continuous iterations of the network, a feature space mapping will eventually be obtained, along with the center a and radius R of the hypersphere.

In the final step, a scoring function is designed as follows:

$$d = \|z_i - a\|^2. \tag{10}$$

Utilize the distance between new data, after being processed by the network model, and the center of the sphere to determine whether it is faulty data and the degree of the fault.

The pseudocode for the second phase algorithm is presented in Table 3, resembling Algorithm 2.

3. Case Study

3.1. Process Description. This case study utilizes data from the gas turbine monitoring system of Unit 3 in a power plant using the GE 9FA GENSET model. The dataset includes records from 133 sensors at 139 h. The sensor parameters originate from the different components of the gas turbine and belong to various hardware management systems. The sensor data include turbine speed, turbine exhaust dispersion, compressor inlet temperature, outlet temperature, atmospheric temperature, and atmospheric pressure. Consequently, they often have different sampling frequencies and representation ranges. Therefore, preprocessing is required to use them as input variables for deep neural networks and other models. This preprocessing ensures that equal importance is assigned to different parameters during the network training. Owing to compressor malfunction necessitating factory maintenance, operational data

Table 2: The backbone of the AE network.

```
Algorithm 1: The algorithmic procedure for the first phase
Input: The training set $\{x_i\}_{i=1}^{N}$, batch size $B$, the learning rate $a$, iteration count $K$
Output: Encoder $\psi(\cdot)$, decoder $\varphi(\cdot)$
Initialization: Encoder $\psi(\cdot)$, decoder $\varphi(\cdot)$, parameter set $\theta_E$ and $\theta_D$
for $k = 1$ to $K$ do:
    Randomly select a mini-batch of data $\{x_i\}_{i=1}^{B}$ from the training set $\{x_i\}_{i=1}^{N}$
    Obtain the reconstructed samples $\{\hat{x}_i\}_{i=1}^{B}$ from the training samples $\{x_i\}_{i=1}^{B}$
    $L_{\theta_E, \theta_E} \Leftarrow \frac{1}{B}\sum_{i=1}^{B}\|x_i - \hat{x}_i\|$
    $\theta_E \Leftarrow \theta_E \alpha \nabla_{\theta_E} L_{\theta_E}$, $\theta_D \Leftarrow \theta_D \alpha \nabla_{\theta_D} L_{\theta_D}$
end for
```

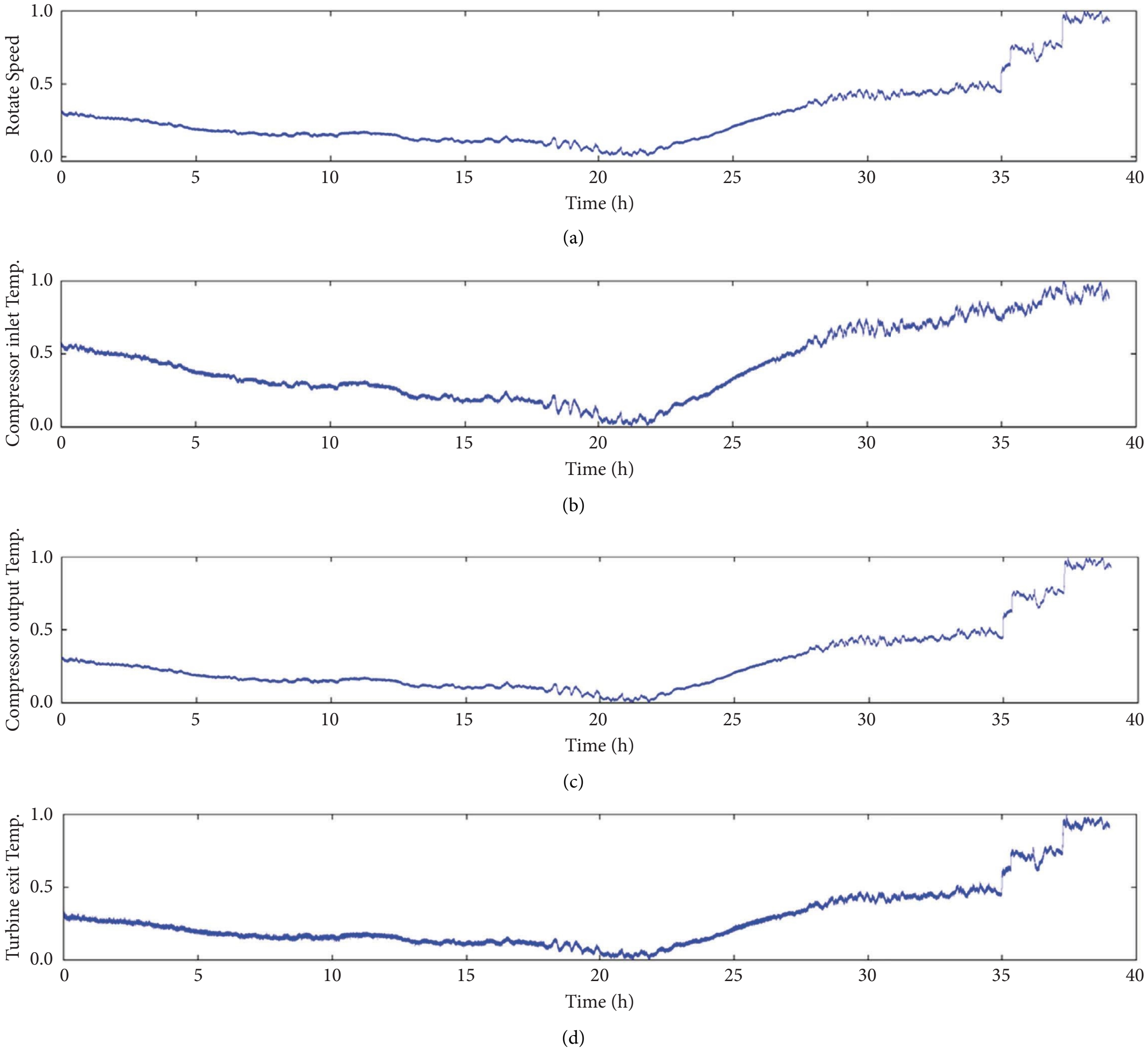
Table 3: Extraction of hidden features and hypersphere optimization.

```
Algorithm 2: The algorithmic procedure for the second phase
Input: The training set $\{x_i\}_{i=1}^{N}$, batch size $B$, the learning rate $a$, iteration count $M$, encoder $\psi(\cdot;W)$, regularization parameter $\lambda$
Output: Encoder $\psi(\cdot;W)$, the center of the hypersphere $a$
Initialization: Encoder $\psi(\cdot;W)$, model parameters $W$
The center $a$ of a hypersphere can be calculated by $a = \frac{1}{n}\sum_{i=1}^{n} z^i$
for $m = 1$ to $M$ do:
    Randomly select a mini-batch of data $\{x_i\}_{i=1}^{B}$ from the training set $\{x_i\}_{i=1}^{N}$
    Randomly select a mini-batch of data
    Input to the encoder $\psi(\cdot;$ W) to obtain the hidden features $Z$
    $L_w \Leftarrow \frac{1}{B}\sum_{i=1}^{B}\|z^i - a\|^2 + \frac{\lambda}{2}\sum_{l=1}^{L}\|w^l\|^2$
    $w \Leftarrow w - \alpha \nabla_w L_w$
end for
The updated center $a$ of the hypersphere is obtained by $a = \frac{1}{n}\sum_{i=1}^{n} z^i$
```

preceding the occurrence of the alarm were gathered from the gas turbine unit for validation purposes. Temporal waveform representations of some parameters, such as the temporal signals of the parameters, including the engine speed, compressor inlet and outlet temperatures, and turbine exit temperature.

The data for 39 h preceding the occurrence of the fault are presented in Figure 8. To validate the feature extraction capabilities of the anomaly state assessment method, monitoring data from the turbine unit were divided into two parts. The initial 100 h of stable and normal operation was chosen as the training set for model training. The 39 h of data monitored before the unit underwent maintenance were utilized for model testing. Because these testing data did not appear during model training, they were more indicative of the model's generalization performance. Within these 39 h of data, the last 4 h exhibited a distinctly abnormal turbine state, allowing this period to be defined as anomalous data, while the preceding 35 h of monitoring data are still designated as normal. This design enabled the evaluation of the model's performance in unforeseen abnormal situations and tested its robustness under normal operating conditions. To facilitate computation, the data were normalized. Blades in compressor stages 10 to 16 incurred varying degrees of damage, along with the impairment of the first-stage nozzles and moving blades of the high-pressure turbine. These parameters, encompassing aspects such as the engine speed and temperatures at the inlet and outlet of the compressor and turbine, provide vital insights into the operational state and performance of the equipment. Temporal signal waveforms provide an intuitive visual display of the operating conditions of the equipment. By scrutinizing features such as waveform shape, amplitude, and frequency, preliminary insights into changing trends and anomalies in the parameters can be obtained. Nevertheless, relying solely on waveform observation often proves insufficient for precisely discerning whether the equipment is undergoing degradation or is facing a malfunction. Consequently, an in-depth analysis necessitates a comprehensive consideration of these waveforms alongside other parameter data to establish more reliable indicators for the assessment of abnormal states.

Considering the substantial number of parameters collected, they can be broadly categorized into two main types: gas path parameters and rotational speed parameters, with gas path parameters encompassing temperature and pressure parameters. In the process of constructing the sample set, for the gas path parameters, random sampling was conducted by unfolding the gas path data in the training set using a sliding window with a length of 200, establishing gas path training samples. Simultaneously, for the rotational speed data, random sampling was performed on the data using a sliding window of the same length, creating rotational speed training samples. The testing samples were obtained by sequentially sampling the test set using a sliding window with a length of 200 and an overlap rate of 0.3. Following the sampling process, the final counts of training and test set samples were 3600 and 2005, respectively. All samples in the training set were healthy, whereas there were 1800 healthy samples and 205 anomalous samples in the test set. To analyze and evaluate the trained network, precision, recall, F1 score, and smoothness were employed as the four



(a)

(b)

(c)

(d)

Figure 8: Temporal waveform: (a) rotate speed of engine; (b) compressor inlet temperature; (c) compressor outlet temperature; (d) turbine exit temperature.

metrics for the model assessment. The first three metrics were utilized to measure the anomaly detection accuracy of the model, while smoothness is employed to evaluate the denoising ability of the model, using the mean absolute deviation. A smaller smoothness value indicates that the model is less influenced by noise.

3.2. Experimental Platform and Hyperparameter Configuration. The network was built on the PyCharm platform using the PyTorch framework. The experiments were conducted on a computer equipped with an RTX 2060 GPU and i5-8400 CPU. The model training utilized the Adam optimizer with an initial learning rate of 0.001, a batch size of 18, 50 iterations for the first stage, and 30 iterations for the second stage.

To prevent overfitting during the training process, L2 regularization, early stopping, and learning rate annealing techniques were employed in the network structure. The introduction of L2 regularization parameters compresses the weight values of the neural network close to zero, reducing the magnitude of parameter changes. In this experiment, L2 was set to 10−5. The learning rate annealing technique involves reducing the learning rate when the model performance ceases to improve significantly after several training rounds. If the loss value did not decrease for three consecutive rounds, the learning rate was halved; for example, automatically adjusting from 0.1 to 0.05. Early stopping aims to halt training when the model performance reaches a plateau, which was triggered if the loss value did not decrease for five consecutive rounds in this experiment.

4. Results and Discussion

4.1. Ablation Experiments. To validate the effectiveness of the DMS-SVDD method, ablation experiments were performed to assess the impact of each component (multiscale convolutional neural network (M) and soft-threshold activation network (S)) on the overall performance of the model. Specifically, we compared the performance of the original Deep-SVDD model with three variants: Deep-SVDD + S (with only the soft-threshold activation network added), Deep-SVDD + M (with only the multiscale convolutional neural network added), and DMS-SVDD (with both M and S added). The comparison was based on metrics such as accuracy, recall, F1 score, and smoothness. These metrics are defined as follows.

Precision: Precision is the ratio of true positive predictions to the total number of positive predictions made. It measures the accuracy of the positive predictions

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}. \tag{11}$$

Recall: Recall, also known as sensitivity, is the ratio of true positive predictions to the total number of actual positives. It measures the ability of the model to capture all relevant instances

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}. \tag{12}$$

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both the precision and recall of a model

$$F1\ \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

Smoothness: Smoothness is not a standard metric like the others and can be defined in various ways depending on the context. In the context of time-series data or signal processing, smoothness could refer to the degree of fluctuation or variation in the signal. One way to quantify smoothness is by calculating the average absolute difference between consecutive data points

$$\text{Smoothness} = \frac{1}{N-1} \sum_{i=1}^{N-1} |x_{i+1} - x_i|, \tag{14}$$

where N is the total number of data points and x_i is the value of the data point at position i. A lower value of smoothness indicates a smoother signal. These formulas can be used to evaluate the performance of classification models and the quality of signals or time-series data.

The confusion matrix for the SVDD network with the added module is shown in Figure 9. The comparison was based on metrics such as accuracy, recall, F1 score, and smoothness, as shown in Table 2.

The analysis in Table 4 reveals that DMS-SVDD achieves a very high accuracy (0.9985), indicating its effectiveness in classifying normal and anomalous instances. F1 reflects a good balance between the precision and recall. A high recall of 0.9989 indicated its ability to identify a high proportion of actual anomalies. Simultaneously, lower smoothness indicates that it defines tight decision boundaries. The DMS-SVDD excels in terms of accuracy, recall, F1 score, and smoothness. Particularly noteworthy is its outstanding performance in smoothness, where it consistently achieved the lowest values among the four comparison sets. This is attributed to the fact that in certain contexts of anomaly detection or signal processing, lower smoothness indicates a better capability to capture sudden changes or anomalies effectively. Compared with Deep-SVDD, DMS-SVDD exhibited an improvement of 21.04% in accuracy, a 24.09% increase in recall, a 12.02% enhancement in F1 score, and a reduction in smoothness by 51.38%. In contrast to OCSVM, DMS- SVDD demonstrates a notable enhancement, with a 47.1% increase in accuracy, a 45.27% increase in recall, a 25.91% improvement in F1 score, and a decrease in smoothness by 52.68%. Relative to AE, DMS-SVDD showed improvements across metrics, including a 22.94% increase in accuracy, a 32% increase in recall, a 16.87% enhancement in F1 score, and a decrease in smoothness by 39.15%.

The changes in the anomaly evaluation metrics, normalized on the test set, for the four methods are shown in Figure 10. The anomaly evaluation metrics include accurate, recall, F1, and smoothness. These metrics were used to assess the model's ability to correctly identify anomalies and distinguish them from normal instances. A comparison indicates that the multiscale module (M) effectively enhances the feature extraction capabilities and improves model recognition accuracy. Meanwhile, the soft-threshold module (S) slightly enhanced the model accuracy but significantly enhanced the denoising ability of the model. The synergistic application of both methods (M and S) in DMS-SVDD achieved optimal performance.

4.2. Model Training Efficiency. In the comparison of model performance, particular attention was given to two key aspects of model training for DMS-SVDD, in contrast to the original Deep-SVDD, Deep-SVDD + S, and Deep-SVDD + M models: convergence epochs and training time. The following is a summary of the convergence epochs and duration of single-round training for each model in each training stage, as presented in Table 5.

By comparing these data sets, it can be inferred that the refined model demonstrates a significant reduction in the number of rounds needed for convergence, despite experiencing an increase in the duration of single-round training at each stage. As a result, the overall training duration paradoxically became shorter, emphasizing improved training efficiency. DMS-SVDD displays notable advantages over the other models in terms of convergence rounds and

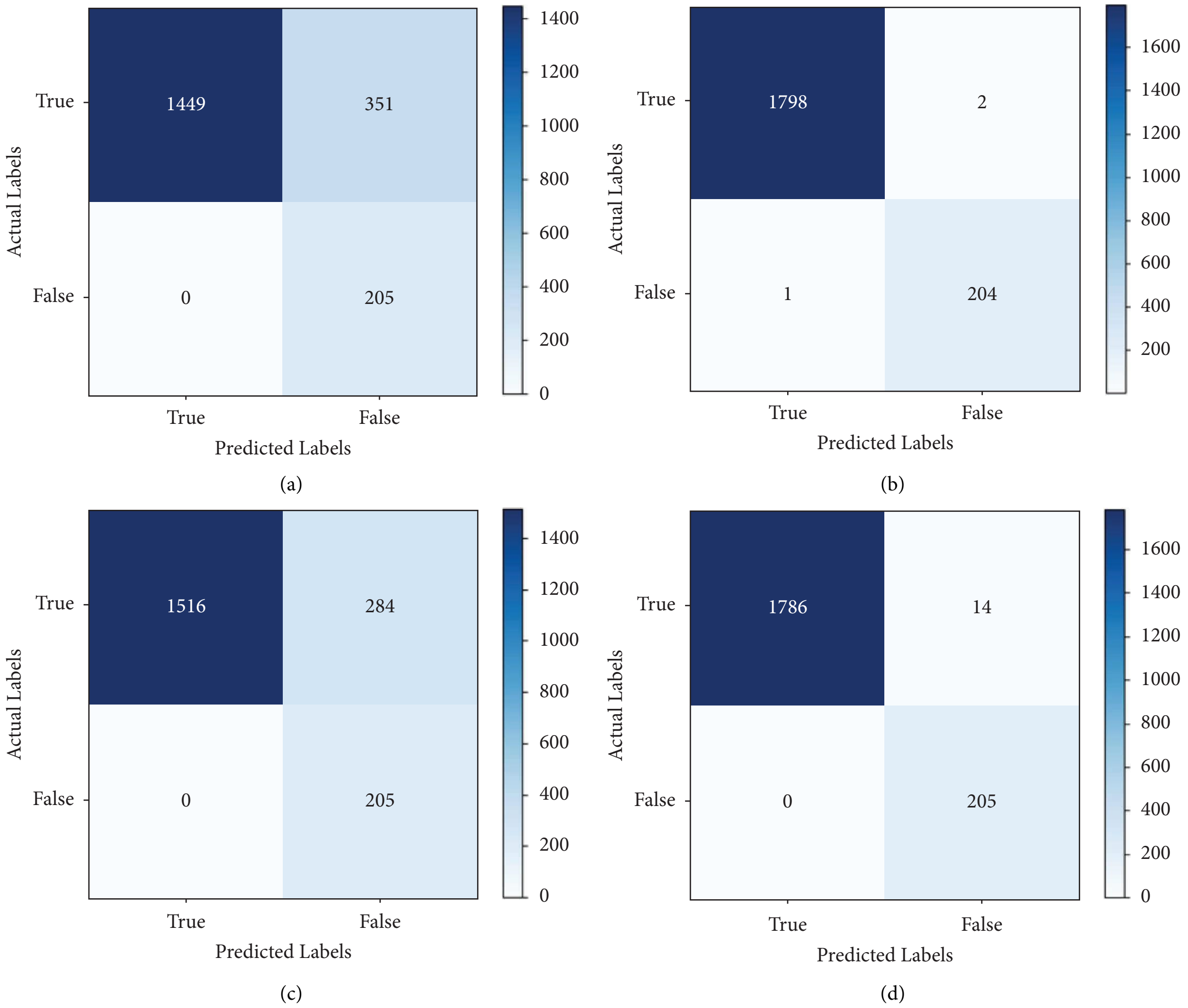


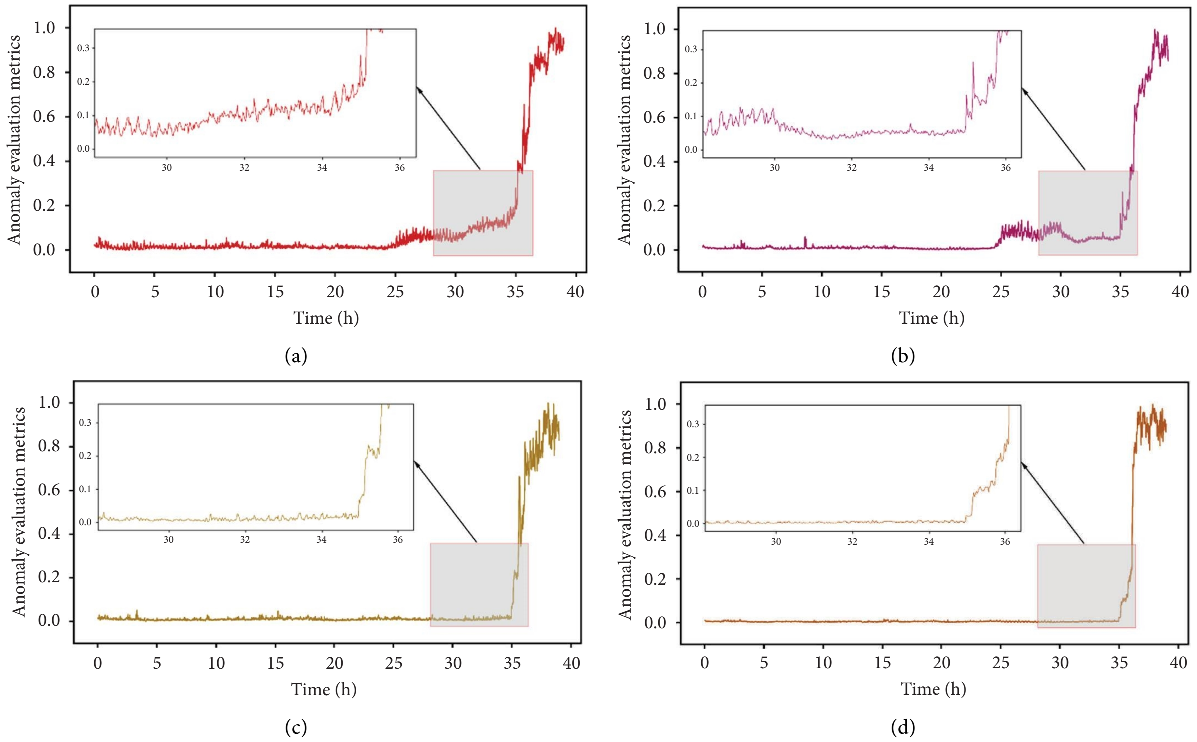
Figure 9: Confusion matrix of the original model: (a) Deep-SVDD; (b) DMS-SVDD; (c) Deep-SVDD + S; (d) Deep-SVDD + M.

Table 4: Comparative performance of DMS-SVDD and other anomaly detection algorithms in ablation experiments.

	Accurate	Recall	F1 score	Smoothness
Deep-SVDD	0.8249	0.8050	0.8920	0.00617
Deep-SVDD + S	0.8584	0.8422	0.9144	0.00402
Deep-SVDD + M	0.9930	0.9922	0.9961	0.00515
OCSVM	0.6788	0.6876	0.7936	0.00634
AE	0.7696	0.7567	0.8550	0.00493
DMS-SVDD	0.9985	0.9989	0.9992	0.00300

training time. In particular, in the initial stage, DMS-SVDD shows a faster convergence rate while remaining competitive in the duration of single-round training in the second stage. This underscores the exceptional performance of DMS-SVDD in enhancing model training efficiency, providing robust support for practical applications where sensitivity to training time is crucial. Compared to Deep-SVDD, DMS-SVDD achieved an overall reduction of 34.13% in the training time.

In the comparative analysis of model training efficiencies, incorporate the computational demands and parameter counts of various models, as illustrated in Table 6. Synthesizing data from Tables 5 and 6 yields the following insights: from the perspective of model complexity, as the complexity increases (through the incorporation of multiscale convolutional neural networks and soft-threshold activation networks), both the computational complexity (FLOPs) and the number of parameters rise. The DMS-SVDD model exhibits the highest values in both computational demand and parameter count in its two phases, indicating its status as the most complex model. Data from Table 6 suggests that despite the increased computational load of the DMS-SVDD model, its total training time is comparatively reduced. This reduction is attributed to the



(a)

(b)

(c)

(d)

Figure 10: Anomaly evaluation metrics: (a) Deep-SVDD; (b) Deep-SVDD + S; (c) Deep-SVDD + M; (d) DMS-SVDD.

Table 5: The time consumed during model training.

	Convergence rounds in the initial stage	Duration of a single-round training in the initial stage(s)	Convergence rounds in the second stage	Duration of a single-round training in the second stage(s)	Total training duration(s)
Deep-SVDD	36	12.7	21	9.2	650.4
Deep-SVDD + S	27	15.4	12	11.8	557.4
Deep-SVDD + M	16	23.5	14	16.9	612.6
DMS-SVDD	12	25.2	7	18.0	428.4

Table 6: Computational resources consumed by model training.

Model	The initial stage		The second stage	
	Computation/FLOPs (M)	Number of parameters (10^6)	Computation/FLOPs (M)	Number of parameters (10^6)
Deep-SVDD	92	8.0	72	6.6
Deep-SVDD + S	142	10.3	102	8.4
Deep-SVDD + M	264	14.7	183	11.4
DMS-SVDD	292	16.3	215	13.9

DMS-SVDD model's enhanced efficiency in feature extraction and denoising, allowing it to converge in fewer training rounds and, thereby, learn data representations more swiftly, reducing the time required for training. Furthermore, the DMS-SVDD model's high precision, recall, and F1 scores in anomaly detection tasks demonstrate its ability to utilize computational resources more effectively during training, leading to performance improvements.

According to Table 5, the DMS-SVDD model converges in 12 rounds in the first phase and 7 rounds in the second phase, with a total training duration shorter than other models. This indicates that the model can reach a convergence state within a relatively small number of training rounds, reducing training time and improving efficiency, meaning the model can learn the data distribution more quickly. The model's high precision, recall, and F1 scores on the test set indicate that, even with fewer convergence rounds, it still maintains good performance. This further confirms the model's convergence; i.e., the model can reach a satisfactory level of performance within a limited training time. Noise can mask or distort the true patterns in data, making it difficult for models to accurately identify anomalies. If a model is very sensitive to noise, its output may become unstable in the presence of noise, affecting the accuracy of anomaly detection. The minimal smoothness of the DMS-SVDD model proposed in this paper means the model's output changes less between consecutive data points, typically indicating that the model can stably handle data, maintaining consistency even in the presence of noise. The DMS-SVDD model exhibits good convergence in terms of training efficiency, model performance, and denoising capability.

4.3. Comparative Analysis of Anomaly Detection Models. To demonstrate the significant advantages of the proposed DMS-SVDD model, a comparison was conducted with two conventional unsupervised learning methods: OCSVM and AE for anomaly detection.

OCSVM is a variant of the traditional support vector machine designed for anomaly detection. After training, the model acquired an optimal hyperplane. The model takes manually engineered features as input, namely, peak-to-peak value, root mean square, standard deviation, and skewness of the segmented raw data from each sensor signal. The optimal parameters of the model were determined through five repeated experiments.

The principle of AE in unsupervised anomaly detection relies on a smaller reconstruction error for normal data and a larger reconstruction error for anomalous data. This is owing to the AE's limited exposure to abnormal data during the training process, resulting in weaker reconstruction capabilities for such data. Consequently, when the input data significantly deviate from the normal patterns, the reconstruction error increases, leading to an anomalous classification. The model structure shares the autoencoding component with Deep-SVDD, with the input as the raw signals from multiple sensors, and the training was conducted over 50 iterations.

In Figure 11, a horizontal black dashed line has been added to represent the threshold for fault detection across all three models. Observations below this line are considered normal, while those above indicate potential faults. From Figure 11, the following conclusions can be drawn:

(i) Trend and Stability of Indicators. Distinct variations were observed among the methods concerning their ability to reflect the trend and stability of anomaly development. The DMS-SVDD model, indicated by the blue line, remains consistently below the threshold, demonstrating its stability and suggesting a high degree of reliability in distinguishing between normal operation and genuine anomalies. The AE model, depicted by the red line, shows a spike crossing the threshold at the 13th hour, which does not align with the actual occurrence of anomalies, indicating potential false positives. The OCSVM model, represented by the green line, exhibits frequent oscillations crossing the threshold, highlighting its potential instability and tendency for false alarms.

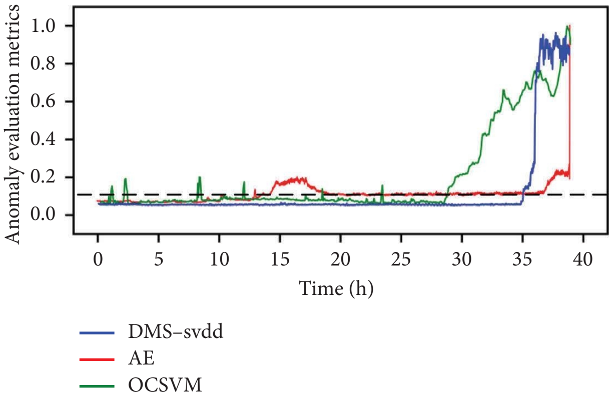


Figure 11: Anomaly evaluation metrics of DMS-SVDD, AE, and OCSVM.

(ii) Early Fault Detection Capability. The DMS-SVDD model's ability to detect early faults is substantiated by its evaluation metrics, which closely align with the actual anomalies observed in the original time-domain signals. This is in stark contrast to the AE model, which shows delayed and less distinct responses, and the OCSVM model, which demonstrates erratic behavior before detecting actual faults.

It is worth noting that the OCSVM model's early indications of a fault, as seen in Figure 11, could be misleading due to its higher sensitivity to noise, resulting in a higher rate of false alarms. The DMS-SVDD model's approach to balance sensitivity with specificity aims to minimize such false positives while maintaining the capability for timely and accurate fault detection.

Through this comparative analysis, the DMS-SVDD model not only effectively integrates anomalous features for the early detection of faults but also demonstrates a clear trend of performance changes without the instability observed in other models. This supports the model's application as a valuable tool for decision-making in the maintenance of combustion engines.

5. Conclusion

In this investigation, we introduced an innovative approach for detecting anomalies and predicting faults in gas turbines by employing a DMS-SVDD. The pivotal elements of our proposed methodology include the application of an SVDD and a Deep Autoencoder Backbone Network Framework.

The primary innovation lies in the integration of a multiscale convolutional neural network (M) and soft-threshold activation network (S) into the Deep-SVDD framework. This fusion enhances the feature extraction capabilities and improves the ability of the model to denoise, resulting in a more robust anomaly detection model. Compared with Deep-SVDD, DMS-SVDD exhibits an improvement of 21.04% in accuracy, a 24.09% increase in recall, a 12.02% enhancement in F1 score, and a reduction in smoothness by 51.38%. In contrast to OCSVM, DMS-SVDD demonstrates a notable enhancement, with a 47.1% increase in accuracy, a 45.27% increase in recall, a 25.91% improvement in F1 score, and a decrease in smoothness by 52.68%. Relative to AE, DMS-SVDD showed improvements across metrics, including a 22.94% increase in accuracy, a 32% increase in recall, a 16.87% enhancement in F1 score, and a decrease in smoothness by 39.15%.

The DMS-SVDD model exhibits exceptional performance, outperforming traditional methods such as OCSVM and AE in terms of accuracy, recall, F1 score, and smoothness. This signifies the ability of the model to accurately identify anomalies and maintain low sensitivity to noise. By effectively capturing subtle changes in operational patterns, the DMS-SVDD proves invaluable in identifying anomalies at an early stage, offering a proactive approach to maintenance.

DMS-SVDD demonstrates improved training efficiency by reducing convergence rounds and overall training times. The DMS-SVDD demonstrated enhanced training efficiency with a reduction in the convergence rounds by 66% and overall training times by 34.13%. This efficiency enhancement is crucial for practical applications, in which timely responses to changing conditions are imperative.

The DMS-SVDD offers a promising solution for gas turbine anomaly detection by combining feature extraction, denoising, and early fault detection capabilities. While the model's performance improvements over existing methods are notable, it is important to acknowledge that the current application has been limited to a restricted number of hours for only one fault type in one machine. Future research could explore further refinements to the DMS-SVDD model and its application to diverse industrial contexts, including more extensive datasets and a broader range of fault types and machines. By expanding the scope of the study, the practical advantages of the DMS-SVDD model can be more thoroughly evaluated, potentially making it a valuable tool for decision support in the maintenance of gas turbine systems and other industrial applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by China Postdoctoral Science Foundation (Grant number: 282205).

References

[1] C.-Z. Dong and F. N. Catbas, "A review of computer vision–based structural health monitoring at local and global levels," *Structural Health Monitoring*, vol. 20, no. 2, pp. 692–743, 2021.

[2] C. Qu, T. Yi, X. Yao, and H. Li, "Complex frequency identification using real modal shapes for a structure with proportional damping," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 10, pp. 1322–1336, 2021.

[3] M. Salkhordeh, M. Mirtaheri, and S. Soroushian, "A decision-tree-based algorithm for identifying the extent of structural damage in braced-frame buildings," *Structural Control and Health Monitoring*, vol. 28, no. 11, 2021.

[4] R. Fakhimi, M. Shahabsafa, W. Lei et al., "Discrete multi-load truss sizing optimization: model analysis and computational experiments," *Optimization and Engineering*, vol. 23, no. 3, pp. 1559–1585, 2022.

[5] G. de Novaes Pires Leite, F. C. Farias, T. G. de Sá et al., "A robust fleet-based anomaly detection framework applied to wind turbine vibration data," *Engineering Applications of Artificial Intelligence*, vol. 126, Article ID 106859, 2023.

[6] K. Hendrickx, W. Meert, Y. Mollet et al., "A general anomaly detection framework for fleet-based condition monitoring of machines," *Mechanical Systems and Signal Processing*, vol. 139, Article ID 106585, 2020.

[7] Z. Chen, Y. Cao, S. X. Ding et al., "A distributed canonical correlation analysis-Based fault detection method for plant-wide process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2710–2720, 2019.

[8] X. Deng, P. Cai, Y. Cao, and P. Wang, "Two-step localized kernel principal component analysis based incipient fault diagnosis for nonlinear industrial processes," *Industrial & Engineering Chemistry Research*, vol. 59, no. 13, pp. 5956–5968, 2020.

[9] P. Cui, C. Zhan, and Y. Yang, "Improved nonlinear process monitoring based on ensemble KPCA with local structure analysis," *Chemical Engineering Research and Design*, vol. 142, pp. 355–368, 2019.

[10] H. Habbak, M. Mahmoud, M. M. Fouda et al., "Efficient one-class false data detector based on deep SVDD for smart grids," *Energies*, vol. 16, no. 20, p. 7069, 2023.

[11] B. C. Liu, H. T. Xue, D. Y. Ding, N. Sun, and P. Chen, "In-wheel motor fault diagnosis using affinity propagation minimum-distance discriminant projection and weibull-kernel-function-based SVDD," *Sensors*, vol. 23, no. 8, p. 4021, 2023.

[12] J. Yang, F. Y. Cheng, M. Duodu, M. Li, and C. Han, "High-precision Fault Detection for electric vehicle battery system based on bayesian optimization SVDD," *Energies*, vol. 15, no. 22, p. 8331, 2022.

[13] Y. H. Liu and Y. J. Chen, "Automatic defect detection for TFT-LCD array process using quasiconformal kernel support vector data description," *International Journal of Molecular Sciences*, vol. 12, no. 9, pp. 5762–5781, 2011.

[14] Y. L. Wan, Y. W. Fei, R. Jin, T. Wu, and X. G. He, "An object-oriented deep multi-sphere support vector data description method for impervious surfaces extraction based on multi-sourced data," *ISPRS International Journal of Geo-Information*, vol. 12, no. 6, p. 219, 2023.

[15] C. Wang, H. L. Liu, C. Li, Y. X. Sun, W. Wang, and B. Wang, "Robust intrusion detection for industrial control systems using improved autoencoder and bayesian Gaussian mixture model," *Mathematics*, vol. 11, no. 9, p. 2048, 2023.

[16] Z. J. Feng, Z. H. Wang, X. Q. Liu, and J. H. Li, "Rolling bearing performance degradation assessment with adaptive sensitive feature selection and multi-strategy optimized SVDD," *Sensors*, vol. 23, no. 3, p. 1110, 2023.

[17] L. L. Fan, J. H. Zhang, W. T. Mao, and F. K. Cao, "Unsupervised anomaly detection for intermittent sequences based on multi-granularity abnormal pattern mining," *Entropy*, vol. 25, no. 1, p. 123, 2023.

[18] J. A. Navarro-Acosta, I. D. García-Calvillo, V. Avalos-Gaytán, and E. Reséndiz-Flores, "Metaheuristics and support vector data description for Fault Detection in industrial processes," *Applied Sciences*, vol. 10, no. 24, p. 9145, 2020.

[19] X. G. Deng and Z. Zhang, "Nonlinear chemical process fault diagnosis using ensemble deep support vector data description," *Sensors*, vol. 20, no. 16, p. 4599, 2020.

[20] Y. Zhao, X. G. Zhang, Z. J. Shang, and Z. Y. Cao, "A novel hybrid method for KPI anomaly detection based on VAE and SVDD," *Symmetry*, vol. 13, no. 11, p. 2104, 2021.

[21] Q. F. Wang, Y. F. Song, H. Li, Y. Shu, and Y. Xiao, "Tribological behavior characterization and Fault Detection of mechanical seals based on face vibration acceleration measurements," *Lubricants*, vol. 11, no. 10, p. 430, 2023.

[22] Z. Wang, S. X. Wang, S. K. Zhang, and J. Zhan, "An expert system based on data mining for a trend diagnosis of process parameters," *Processes*, vol. 11, no. 12, p. 3311, 2023.

[23] B. D. Oh, H. Choi, W. J. Chin, C. Y. Park, and Y. S. Kim, "Void detection inside duct of prestressed concrete bridges based on deep support vector data description," *Applied Sciences*, vol. 13, no. 10, p. 5981, 2023.

[24] K. Rama Krishna and K. Ramachandran, "Machinery bearing fault diagnosis using variational mode decomposition and support vector machine as a classifier," *IOP Conference Series: Materials Science and Engineering*, vol. 310, Article ID 012076, 2018.

[25] G. Xie, Y. Jiang, and N. Chen, "A multi-class support vector data description approach for classification of medical image," in *Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security*, pp. 115–119, Emeishan, China, December 2013.

[26] Y. C. Ding, K. Ma, T. J. Pu, X. Y. Wang, R. Li, and D. X. Zhang, "A deep learning-based classification scheme for false data injection attack detection in power system," *Electronics*, vol. 10, no. 12, p. 1459, 2021.

[27] S. B. Ji, S. K. Han, and J. T. Rhee, "Multi-view masked autoencoder for general image representation," *Applied Sciences*, vol. 13, no. 22, Article ID 12413, 2023.

[28] M. Sidulova and C. H. Park, "Conditional variational autoencoder for functional connectivity analysis of autism spectrum disorder functional magnetic resonance imaging data: a comparative study," *Bioengineering*, vol. 10, no. 10, p. 1209, 2023.

[29] G. Liang, Z. L. Ji, Q. H. Zhong, Y. Huang, and K. Han, "Vector quantized variational autoencoder-based compressive sampling method for time series in structural health monitoring," *Sustainability*, vol. 15, no. 20, Article ID 14868, 2023.

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

[31] W. D. Xu, J. K. He, W. H. Li et al., "Long-short-term-memory-based deep stacked sequence-to-sequence autoencoder for health prediction of industrial workers in closed environments based on wearable devices," *Sensors*, vol. 23, no. 18, p. 7874, 2023.

[32] Z. Ai, H. Cao, J. Wang, Z. Cui, L. Wang, and K. Jiang, "Research method for ship engine fault diagnosis based on multi-head graph attention feature fusion," *Applied Sciences*, vol. 13, no. 22, Article ID 12421, 2023.

[33] S. Ali and M. Patterson, "Improving isomap efficiency with rks: a comparative study with t-distributed stochastic neighbor embedding on protein sequences," *D-J Series*, vol. 6, no. 4, pp. 579–591, 2023.

[34] Z. G. Liang, L. J. Zhang, and X. Z. Wang, "A novel intelligent method for fault diagnosis of steam turbines based on T-SNE and XGBoost," *Algorithms*, vol. 16, no. 2, p. 98, 2023.

[35] Z. H. Zhang, Y. P. He, and Z. B. Zhang, "Micromanipulation and automatic data analysis to determine the mechanical strength of microparticles," *Micromachines*, vol. 13, no. 5, p. 751, 2022.

[36] S. Shojae Chaeikar, A. A. Manaf, A. A. Alarood, and M. Zamani, "PFW: polygonal fuzzy weighted—an SVM kernel for the classification of overlapping data groups," *Electronics*, vol. 9, no. 4, p. 615, 2020.

[37] L. Fang, H. Li, and S. Wang, "An improved soft thresholding method for image denoising using deep learning," *IEEE Access*, vol. 9, pp. 42969–42978, 2021.

[38] J. Zeng, S. Lin, Y. Wang, and Z. Xu, "A global-local optimization framework for image deconvolution with soft thresholding prior," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3217–3230, 2019.

[39] H. Zhang, J. Wu, and B. Li, "A soft thresholding-based deep neural network for DOA estimation," *IEEE Access*, vol. 9, pp. 66783–66793, 2021.

[40] Z. Liu and T. Zeng, "A generic framework of soft-thresholding operators for nonsmooth nonconvex optimization," *Journal of Global Optimization*, vol. 77, no. 3, pp. 525–548, 2020.

[41] C. Li, W. Yin, and Y. Zhang, "User's guide to SURE guided soft thresholding," *Journal of Computational & Graphical Statistics*, vol. 29, no. 2, pp. 269–285, 2020.

[42] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[43] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[44] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.