Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060564

ARTICLE

Robust Deep One-Class Classification Time Series Anomaly Detection

Zhengdao Yang[1], Xuewei Wang[2], Yuling Chen[1,*], Hui Dou[1] and Haiwei Sang[3]

[1]State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, 550000, China
[2]College of Computer Science and Technology, Weifang University of Science and Technology, Weifang, 261000, China
[3]School of Mathematics and Big Data, Guizhou Education University, Guiyang, 550018, China

*Corresponding Author: Yuling Chen. Email: ylchen3@gzu.edu.cn

Received: 04 November 2024; Accepted: 04 March 2025; Published: 19 May 2025

ABSTRACT: Anomaly detection (AD) in time series data is widely applied across various industries for monitoring and security applications, emerging as a key research focus within the field of deep learning. While many methods based on different normality assumptions perform well in specific scenarios, they often neglected the overall normality issue. Some feature extraction methods incorporate pre-training processes but they may not be suitable for time series anomaly detection, leading to decreased performance. Additionally, real-world time series samples are rarely free from noise, making them susceptible to outliers, which further impacts detection accuracy. To address these challenges, we propose a novel anomaly detection method called Robust One-Class Classification Detection (ROC). This approach utilizes an autoencoder (AE) to learn features while constraining the context vectors from the AE within a sufficiently small hypersphere, akin to One-Class Classification (OC) methods. By simultaneously optimizing two hypothetical objective functions, ROC captures various aspects of normality. We categorize the input raw time series into clean and outlier sequences, reducing the impact of outliers on compressed feature representation. Experimental results on public datasets indicate that our approach outperforms existing baseline methods and substantially improves model robustness.

KEYWORDS: Time series anomaly detection; self-supervised learning; robustness

1 Introduction

Analyzing time series allows us to understand the underlying processes that generate these sequences, thereby enhancing our comprehension of these processes. Anomaly detection in time series is a critical challenge in data mining, with applications across diverse fields such as transportation and manufacturing, where it serves to monitor system behavior. For instance, in aircraft engine fault detection, monitoring the time series of engine RPM, fuel pressure, temperature, and vibration signals enables the identification of abnormal fluctuations or patterns, allowing for early detection of mechanical failures and preventing accidents during flight. In railway transportation systems, monitoring the status of trains involves real-time analysis of data such as wheel temperature, axle vibration, and electrical signals. By detecting anomalies, potential hazards that could lead to derailments or failures can be identified and mitigated promptly. In manufacturing production lines, equipment health management is facilitated through the monitoring of vibration, temperature, and current of machinery via time series data, enabling the identification of potential failures and timely maintenance, thus minimizing downtime and ensuring production efficiency.

Time Series Anomaly Detection (TSAD) is also essential in applications like health monitoring and fraud detection, where it involves identifying unique time series instances that deviate from typical

patterns [1]. For example, in health monitoring, detecting abnormal heart rates and brain waves in electrocardiogram (ECG) and electroencephalogram (EEG) data can reveal health issues such as arrhythmias and epilepsy. In credit card fraud detection, analyzing user transaction patterns and identifying unusual behaviors, such as large purchases in a short time or transactions at atypical locations, helps in detecting and preventing fraudulent activities. Additionally, in network traffic anomaly detection, analyzing time series data of network traffic can uncover abnormal data transmission behaviors, such as DDoS attacks, data breaches, or malicious software activities.

In recent years, deep learning-based approaches have achieved impressive results in time series anomaly detection, particularly with complex datasets. These methods excel at modeling long-term and nonlinear temporal patterns within the data, surpassing traditional methods such as similarity search and density-based clustering [2–5]. Neural network-based approaches commonly use an encoder to compress the input time series into a compact latent representation, which is then decoded to reconstruct the original series. This encoder-decoder framework, commonly referred to as an autoencoder [6], compresses the initial input through a bottleneck layer, promoting the formation of compact latent representations. This structure enables the model to capture essential patterns within the time series while filtering out irrelevant or atypical patterns, such as anomalies [7]. This approach aids in the detection of anomalies by evaluating the reconstruction error between the original time series and its reconstructed counterpart. A larger reconstruction error suggests an increased probability that the related observations are anomalies. One-class (OC) techniques consolidate normal instances into a single category by minimizing the volume of the hypersphere that encompasses the feature representations. However, despite the strong performance demonstrated by deep learning techniques [8–10] such as autoencoders, they encounter two major challenges.

Single Hypothesis: A single hypothesis often captures only a limited aspect of sample normality, while anomalies can manifest in diverse forms, such as point anomalies, subsequence anomalies, and anomalies across entire time series. Inspired by the principles of ensemble anomaly detection methods, we suggest that detectors relying on a singular hypothesis may be insufficient to capture this diversity, potentially limiting their effectiveness in detecting varied anomaly types [11]. Consequently, while these methods may excel in detecting specific types of anomalies, their effectiveness often diminishes when encountering others.

An alternative approach divides the process into two phases: pre-training on the overall time series, followed by fine-tuning for anomaly detection (AD). For instance, deep SVDD [12] initially utilizes an autoencoder for feature extraction and subsequently refines these features for anomaly detection through a one-class loss function. Similarly, Reference [13] applies contrastive learning in the first phase and utilize one-class methods for detection in the second phase. Formally, these approaches combine feature extraction methods with the assumptions of one-class learning; however, the objectives of the two phases are often misaligned. While the pre-training phase may yield representations that align with typical patterns, these representations can also be influenced by extraneous features unrelated to anomaly detection. As a result, the performance of such methods may be constrained by these pre-trained features.

Robustness: In unsupervised learning, training data often includes anomalies. Because the encoder compresses all observations in the input time series, even those that are anomalous, the hidden representation may become susceptible to these outliers. Particularly when anomalies are high in amplitude, even a few can compromise the latent information, leading to a risk that the latent representation itself reflects anomalous patterns from the training data.

Deep learning-based soft sensors for time series anomaly detection exhibit significant vulnerabilities to adversarial attacks. Knowledge-guided adversarial perturbations can be designed to subtly manipulate the input data distribution without causing noticeable changes, resulting in a substantial degradation of model performance. This exposes the insufficient robustness of current deep learning methods in time

series anomaly detection [14]. Reference [15] proposes an adversarial training strategy using historical gradients and domain adaptation. By leveraging historical information to capture temporal dynamics and mapping input samples to a shared feature space, this approach effectively enhances model robustness against adversarial examples. This is particularly critical for time series anomaly detection, where data distribution changes introduce additional uncertainty. As a result, the model may produce lower reconstruction errors for specific anomalies, making it difficult to distinguish them from normal data, which in turn negatively affects accuracy. As shown in Fig. 1, training set data contaminated by anomalies can lead to the model generating smaller reconstruction errors for similar anomalies during testing, complicating their detection. To mitigate this issue, a robust solution is required to ensure that the latent representation is not influenced by anomalies present in the training data.

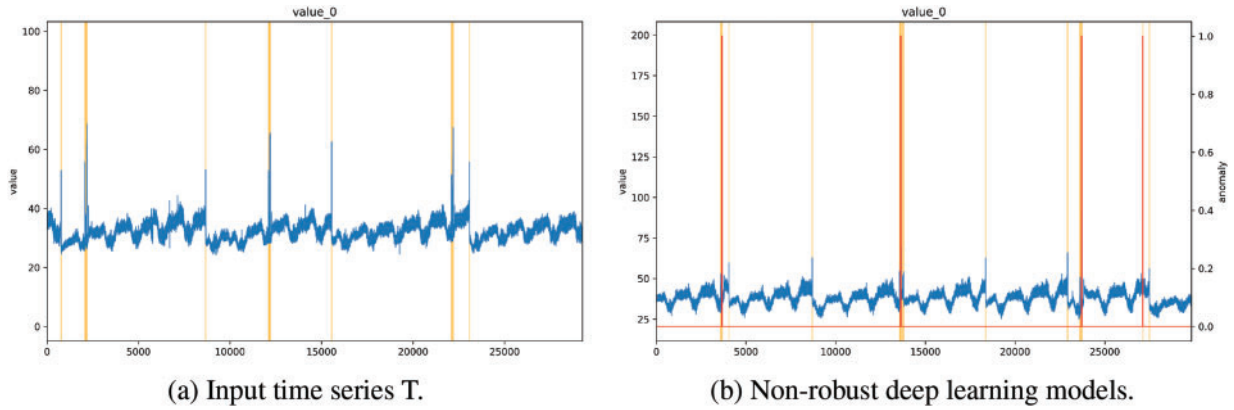

Figure 1: A portion of the time series from the training set for anomaly detection, where the orange shading represents the actual anomalous segments and the red solid line represents the anomaly score. In unsupervised anomaly detection, the training set typically relies on benign samples, which results in the model learning characteristics associated with anomalies. Consequently, the model becomes less sensitive to certain anomalies present in the test set

We introduce a novel autoencoder framework to address the issues of robustness and single hypothesis in anomaly detection. In this study, we introduce a single-stage anomaly detection approach termed the Robust One-Class Classification (ROC) autoencoder. We hypothesize that normal samples will be more accurately reconstructed, with their projection vectors in the latent space forming a compact hypersphere.

Rather than reconstructing the input time series T directly, we decompose it into two components: the clean time series T_L and the anomalous time series T_s. Using an integrated autoencoder, we then reconstruct only the clean component T_L, thereby ensuring that the latent representation remains unaffected by anomalous data. This approach enhances robustness and, consequently, improves the model's accuracy.

In summary, the main contributions of this paper are as follows:

- The proposed method differs from traditional single-hypothesis autoencoder approaches by integrating a multi-hypothesis anomaly detection framework that combines autoencoders with OC methods, capturing richer feature representations in the time series.
- Unlike standard denoising autoencoders, our method does not require additional noise-free training data. By optimizing the sparsity of T_s in the objective function, we can effectively separate the anomalous components in the training data, achieving the effect of noise-free training data.
- The effectiveness of the proposed approach has been validated on publicly available time series datasets, where it demonstrates superior performance compared to existing methods.

2 Related Work

2.1 Problem Definition

A time series $T = \langle s_1, \ldots, s_C \rangle$ consists of C observations, with each observation s_i in $\mathbb{R}^D$. If $D = 1$, then T is a univariate time series; if $D > 1$, it represents a multivariate or multidimensional time series.

For a specified time series $T = \langle s_1, \ldots, s_C \rangle$, we aim to calculate the anomaly score $OS(s_i)$ for every observation s_i. Observations with higher anomaly scores are more likely to be classified as anomalies. Anomalies are not specifically categorized as point anomalies or sequence anomalies; rather, a set of consecutive observations is considered a sequence anomaly if they share high anomaly scores.

2.2 Deep Learning-Based Anomaly Detection

Traditional anomaly detection methods rely on statistical features, but these approaches are often unsuitable for time series data, leads to the feature information not being correctly represented. With the increase in data dimensionality and volume, deep learning methods have emerged. OC methods can capture complex features that represent "normality." For instance, methods based on Generative Adversarial Networks (GANs) [16] and autoencoders [17] assume that normal samples can be reconstructed better by the model. Clustering methods, on the other hand, posit that normal samples cluster together as a large group, while outlier data points are classified as anomalies [18]. In contrast, contrastive learning approaches enhance the data by making positive samples closer to each other while causing the negative samples to be far apart from each other [19]. However, these assumptions can often be overly simplistic or effective only for specific types of anomalies. Additionally, another class of deep learning methods, such as Deep SVDD, employs a two-stage OC classifier. This involves feature extraction using a pre-trained encoder model, which is then classified using OC-SVM [12]. Nevertheless, this approach tends to separate the training objective from the downstream task, hindering the effective learning of diverse time series features.

2.3 Robust Principal Component Analysis

Given a matrix M, Principal Component Analysis (PCA) is capable of discovering a low-rank matrix that serves as an approximation of M. However, as PCA generally employs Singular Value Decomposition (SVD) to determine the low-rank matrix, it exhibits similar sensitivity to outliers as that observed in SVD. To enhance the effectiveness of Principal Component Analysis (PCA) in the presence of outliers, Robust Principal Component Analysis (RPCA) [20] has been introduced. The aim of RPCA is to decompose the matrix M into two components: a low-rank matrix L that represents the underlying clean data and a sparse matrix S that captures the outliers. Specifically, RPCA expresses the original matrix M as follows:

$$M = L + S \tag{1}$$

In this decomposition, L serves as a low-rank matrix that approximates the clean data within the original matrix M, while S represents a sparse matrix composed of elements identified as outliers, which are not encapsulated by the low-rank matrix L. RPCA accomplishes this decomposition by resolving the optimization problem outlined in the following formula.

$$\underset{L,S}{\operatorname{argmin}} \quad \operatorname{rank}(L) + \lambda \|S\|_0 \quad \text{s.t.} \quad X = L + S \tag{2}$$

In this context, $\operatorname{rank}(L)$ represents the rank of matrix L; $\|S\|_0$ is the ℓ_0 norm of matrix S, which indicates the count of non-zero elements within S; and λ serves as a parameter that adjusts the relative significance of $\|S\|_0$. Additionally, given that M is expressed as the sum of L and S, the optimization is subject to the

constraint $M = L + S$. To discover a low-rank matrix L that closely approximates the original matrix M and a sparse matrix S capturing the outliers, the loss function is minimized. Although RPCA is efficient in detecting and excluding outliers, it is limited by its lack of support for time series data and its restriction to linear transformations. Time series data, however, frequently involves complex, nonlinear variations that this approach cannot fully address.

3 Methodology

3.1 Overall Approach

In unsupervised learning, training data may already contain anomalies. As the encoder compresses the time series, the hidden representation becomes highly sensitive to these anomalies. This means that the anomalous information in the training data can contaminate the latent representation, causing the model to learn abnormal features. As a result, the model may exhibit low reconstruction errors for anomalous samples, making it difficult to distinguish them from clean samples.

To address this issue, we propose a robust method that ensures the latent representation is not affected by anomalies in the training data. Drawing on the principles of RPCA, our proposed neural network is designed to partition the input data during training into two segments: the clean time series T_L and the anomalous component T_s, ensuring that $T = T_L + T_s$. We then input the clean series T_L into a deep one-class (OC) method to capture the overall features of the positive samples under multiple hypotheses. The specific approach is illustrated in Fig. 2.

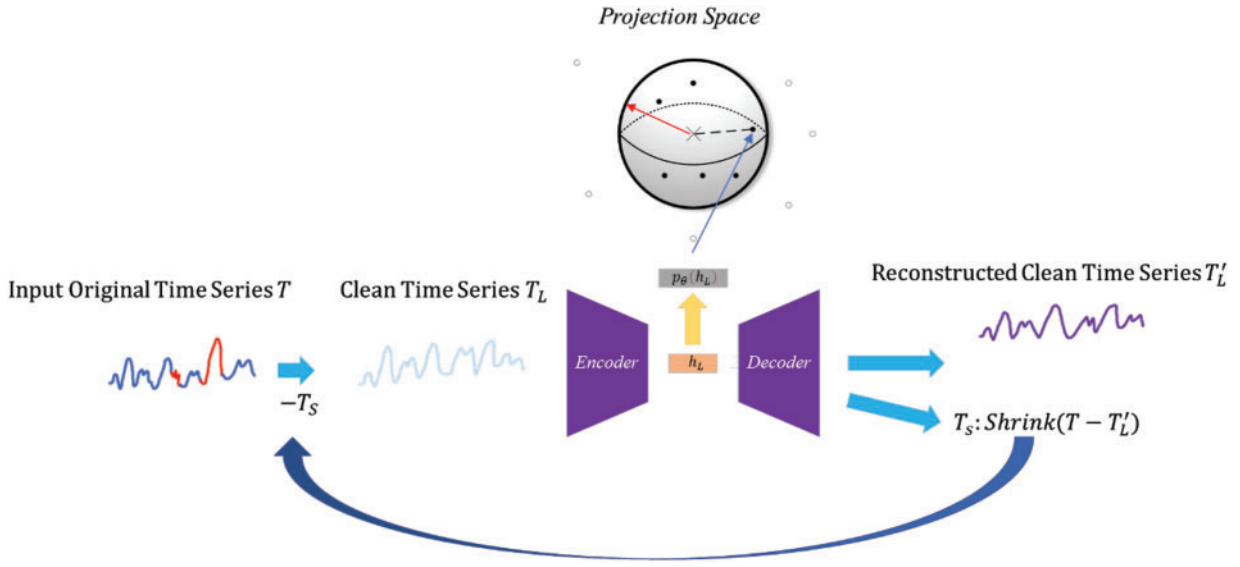

Figure 2: The architecture of the proposed ROC model is depicted, where c represents the center of the hypersphere. Projection vectors of normal instances (e.g., q_i) are contained within the hypersphere, while those of anomalous instances are positioned beyond its boundary. Simultaneously, we separate T_s from T_L, and only clean time series are used as input to the autoencoder, resulting in the reconstruction of the time series T_L'

3.2 Deep OC

In our proposed deep one-class (OC) method, we combine two assumptions: that normal samples can be better reconstructed and that when mapped to a high-dimensional space, they will form a hypersphere with a smaller radius. This approach allows for a more comprehensive learning of the features of clean samples.

The motivation behind this is that under a single assumption, the normality features learned by the model may be one-sided, leading to a situation where certain features of abnormal samples differ from those of normal samples, but those features are not recognized by the model. As a result, specific types of anomalies may go undetected by a model based on a single assumption. The specific objective function is as follows:

$$L = l_{ae}(T, T') + \lambda_1 \cdot l_{OC}(q_i, c) \tag{3}$$

For a given set of N time series training samples, the objective function consists of two parts: one for learning the features of the time series from the perspective of the autoencoder and the other from the one-class (OC) method. The parameter λ_1 controls the weights of these two components. $l_{ae}(T, T')$ represents the reconstruction error of the seq2seq model, defined as follows:

$$l_{\text{ae}}(T, T') = \left\| T - D_{\theta_{\text{AE}}}(E_{\theta_{\text{AE}}}(T)) \right\|_2^2 \tag{4}$$

T' is the reconstructed time series, defined as $D_{\theta_{\text{AE}}}(E_{\theta_{\text{AE}}}(T))$. This is primarily calculated by measuring the mean squared error between the original sequence and the reconstructed sequence, ensuring the model's reconstruction is as close to the original input as possible.

The $l_{OC}(q_i, c)$ represents the OC error defined as:

$$l_{oc}(q_i, c) = \left\| q_i - c \right\|^2 \tag{5}$$

The data point q is obtained by projecting the hidden representations of the training samples from the encoder into a high-dimensional feature space. The distance between q and the center point c is then calculated, with the objective of minimizing the size of the hypersphere centered at c with q as the radius. The center point is determined using a Gaussian mixture model, which can effectively model complex data distributions and handle noise and outliers more effectively.

$$c = \frac{\sum_{i=1}^{N} G(q_i, c) \cdot q_i}{\sum_{i=1}^{N} G(q_i, c)} \tag{6}$$

where $G(q_i, c)$ is the weight function based on the Gaussian distribution, specifically defined as:

$$G(q_i, c) = \exp\left(-\frac{\left\| q_i - c \right\|^2}{2\sigma^2} \right) \tag{7}$$

where σ is the standard deviation of the Gaussian distribution, controlling the degree of fuzziness, and in the testing phase, the classification of the time series T as anomalous is based on the calculated anomaly score S.

$$S = l_{ae}(T, T') + \lambda_1 \cdot l_{oc}(q_i, c) \tag{8}$$

$$x = \begin{cases} \text{anomaly}, & S > \tau \\ \text{normal}, & S \leq \tau \end{cases} \tag{9}$$

where τ is the predefined classification threshold.

3.3 Robust Anomaly Detection

In Principal Component Analysis (PCA), a given matrix M can be approximated by identifying a low-rank matrix. To obtain a low-rank representation, PCA applies Singular Value Decomposition (SVD), which

makes it inherently sensitive to outliers. To enhance robustness in the presence of outliers, Robust Principal Component Analysis (RPCA) has been proposed. RPCA aims to break down the original matrix M into two parts: a low-rank matrix L that represents the underlying clean structure of M and a sparse matrix S that contains the elements identified as anomalies.

Inspired by the approach of RPCA, we can separate the anomalous parts from the input time series and focus solely on learning the benign features from the samples. In this context, the clean time series T_L encapsulates the trends and periodic patterns present in the time series data, whereas T_s identifies the anomalous characteristics, which largely include random fluctuations that do not conform to established patterns. By eliminating this component's influence on the hidden representations, we can more accurately learn the information from clean samples and better differentiate them from anomalous samples. The objective function is as follows:

$$\arg\min \left\| T_L - D_{\theta_{AE}}(E_{\theta_{AE}}(T_L)) \right\|_2 + \lambda_1 \cdot l_{oc}(q, c) + \lambda_2 \cdot \| T_s \|_0 \quad \text{s.t.} \quad T = T_L + T_s \tag{10}$$

$E_{\theta_{AE}}$ represents the encoder part, while $D_{\theta_{AE}}$ represents the decoder part. λ_2 are parameters used to control the balance between the sparsity of T_s. From our analysis, we observe that λ_2 plays a crucial role in separating the anomalous values in the time series. Specifically, when λ_2 is small, the objective function encourages more data to be classified as anomalous and separated from the original data. Conversely, when λ_2 is large, most of the data is retained, with only a small portion being isolated.

In Eqs. (1) and (9), the loss functions include an l_0 norm term to optimize the sparsity of anomalous values while ensuring their semantics. However, the l_0 norm is non-convex, making optimization challenging. According to the [21], transforming the l_0 norm to the l_1 norm can provide a good approximation of the l_0 norm. The formula is as follows:

$$\arg\min \left\| T_L - D_{\theta_{AE}}(E_{\theta_{AE}}(T_L)) \right\|_2 + \lambda_1 \cdot l_{oc}(q, c) + \lambda_2 \cdot \| T_s \|_1 \quad \text{s.t.} \quad T = T_L + T_s \tag{11}$$

3.4 Algorithm

The optimization problems of ROC have constraints and thus cannot be solved by gradient descent based back-propagation (BACKPROP). The optimization task may instead be reformulated into two segments and approached using the Alternating Direction Method of Multipliers (ADMM). ADMM fundamentally works by breaking down the main objective into several sub-objectives, enabling the iterative optimization of each sub-objective while holding the remaining ones constant. Upon optimizing a given sub-objective, the method applies constraints to ensure consistency with the overall objective [22]. Furthermore, the Proximal Algorithm [23] is employed to address elements involving the l_1 norm.

As shown in Algorithm 1, when optimizing the ROC, the process is as follows: first, optimize the integrated autoencoder part by minimizing $L = \lambda_1 \cdot l_{ae}(T, T') + \lambda_2 \cdot l_{OC}(q_i, c)$; then minimize $\| T_s \|_0$; lastly, update $T_L = T - T_s$ to maintain the constraint and provide the result as input for the subsequent iteration. The optimization process concludes based on two criteria: first, when $T = T_L + T_s$ holds, and second, when both T_L and T_s remain constant, indicating that neither T_s nor T_L is further changing, signifying that the anomalous values in T_s have stabilized at an optimal state.

Algorithm 1: Training

Input: Time series T, double λ, double ε
Output: T_L, T_s
1. **Initialization:**

(Continued)

Algorithm 1 (continued)

2. $T_L \leftarrow 0$; $T_s \leftarrow 0$; $T^* \leftarrow T$;
3. **repeat**
4. $T_L \leftarrow T - T_s$;
5. // Optimize AE θ_{AE}
6. Update θ by minimizing $L = \lambda_1 \cdot l_{ae}(T, T') + \lambda_2 \cdot l_{OC}(q_i, c)$; using BACKPROP;
7. $T_L \leftarrow D_{\theta_{AE}}(E_{\theta_{AE}}(T_L))$;
8. $T_s \leftarrow T - T_L$;
9. // Optimize T_s
10. Update T_s by minimizing $\|T_s\|_1$ using PROX;
11. // Compute stopping conditions:
12. condition1 $\leftarrow \frac{\|T - T_L - T_s\|_2}{\|T\|_2}$;
13. condition2 $\leftarrow \frac{\|T^* - T_L - T_s\|_2}{\|T\|_2}$;
14. $T^* \leftarrow T_L + T_s$;
15. **untill** *condition* 1 $<$ ε_1 *or condition* 2 $<$ ε_2
16. **Return:** T_L, T_s

4 Experiments

4.1 Experimental Setup

In terms of datasets, the AIOps dataset is a collection of 29 sub-datasets designed for detecting anomalies in web services based on business cloud KPIs. It includes 29 KPI time series collected from several large technology companies (such as Alibaba, Sogou, Tencent, Baidu, and eBay). These time series are sampled at 1 or 5-min intervals and divided into training and testing portions.

Another dataset we use the UCR Time Series Anomaly Archive, a recently launched repository containing 250 different time series datasets specifically for time series anomaly detection research. Each dataset contains anomalous events of varying lengths, ranging from 1 to 1700. Furthermore, these datasets cover various fields such as health, industry, and biology, exhibiting different types of anomalies with specific characteristics [24].

We also use the multivariate time series dataset SMAP, which comes from a real-world expert-labeled dataset provided by NASA. Each dataset includes a training set and a testing set, with anomalies labeled in the testing set. It consists of data from 27 entities, each monitored by 55 metrics (variables). In all datasets, both point anomalies and collective anomalies are present, and true anomaly labels are available. Moreover, all methods are trained using time series data that contains anomalies, as the datasets do not provide clean time series without anomalies for training purposes. This configuration enables an investigation into the robustness of various algorithms when confronted with anomalies.

Table 1 provides a systematic comparison of the key characteristics of the three datasets (AIOps, UCR, and SMAP) used in this study. It outlines the configurations of the sliding window parameters (window size and time step), the total number of samples, the data splits across training, validation, and testing sets, as well as the anomaly proportions in the training and testing datasets. Notably, the anomaly proportions vary significantly across datasets, with AIOps containing a small proportion of anomalies in both training and testing sets, UCR having no anomalies in the training set and a low proportion in the testing set, and SMAP featuring no anomalies in the training set but a relatively higher anomaly proportion in the testing set. This highlights the diverse nature of the datasets and their suitability for different anomaly detection tasks.

Table 1: Details of dataset

	AIOps	UCR	SMAP
Window size	16	64	64
Time step	2	4	2
Total sample	2,961,039	4,830,858	281,400
Training/validation/testing	40%/10%/50%	24%/6%/70%	32%/8%/60%
Training/testing anomaly	2.98%/1.92%	0%/0.71%	0%/12.79%

Regarding baseline methods, we select two shallow machine learning methods, including OC-SVM [25] and Random Cut Forest (RCF) [26]. In deep learning algorithms, we compare our approach with five algorithms, including the deep one-class method SVDD [12], context-based anomaly detection for time series (TS-TCC) [27], and Ensemble Detection Method AOC [28]. Lastly, we chose two variants of the ROC method for ablation experiments: NoOC, which sets λ_1 to 0, representing a single hypothesis anomaly detection method; and NoRPCA, which directly uses the raw training data without any processing.

In our experiments, we primarily employed **PA**, **PW**, and **Affiliation (precision recall and F1-score)** [29] as evaluation metrics, as they align well with the unique requirements of time series anomaly detection tasks.

PA measures the ratio of correctly classified points (both normal and anomalous) to the total number of points in the time series, offering a global perspective on the model's overall performance. Its formula is as follows:

$$PA = \frac{\text{Number of Correctly Classified Points}}{\text{Total Number of Points}}$$

PA is particularly suitable for scenarios where the primary goal is to assess the model's general classification accuracy across both normal and anomalous data. However, it may have limitations in datasets where normal points significantly outnumber anomalies, as the metric can dilute the model's anomaly detection performance by emphasizing overall accuracy.

PW on the other hand, is designed to focus specifically on anomaly detection by emphasizing the precision and recall of the model when identifying anomalies. It provides a more refined measure of the model's effectiveness in distinguishing anomalous data from normal data [30,31]. The formula for PW Precision is:

$$PW \text{ Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

PW is particularly well-suited for time series anomaly detection tasks where the primary focus is on ensuring the accurate identification of anomalous samples. This metric is valuable in applications such as fault detection in industrial systems, where missing an anomaly (false negative) or misclassifying a normal event (false positive) can lead to significant consequences.

The choice of PA and PW as evaluation metrics reflects their ability to complement each other in time series anomaly detection scenarios. PA offers a holistic view of the model's classification accuracy, while PW ensures the model's stability and effectiveness in specifically detecting anomalies. This dual perspective allows for a balanced evaluation of the model's performance in time series tasks, particularly when addressing real-world applications with imbalanced data distributions or critical anomaly detection requirements.

By integrating these metrics, we can better assess the trade-offs between general accuracy and the precision-recall balance in anomaly detection, ensuring the model's applicability across diverse time series datasets and real-world tasks.

4.2 Implementation Details

In the ROC framework, we utilize two identical three-layer LSTMs (with a dropout rate of 0.45) as Seq2Seq autoencoders. The Adam optimizer was employed with a learning rate of 3×10^{-4}, weight decay of 5×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\varepsilon = 1 \times 10^{-8}$. All methods were implemented using Python 3.9, with PyTorch 1.7 for all neural network-based approaches. Additionally, Sklearn 0.24 was used for OCSVM, while Numpy 1.19 was used for EMA, SSA, and MP. Finally, Statsmodels 0.13 was utilized for STL. All experiments were conducted on a Linux workstation equipped with an Intel 32-core CPU, 256 GB RAM, and a single NVIDIA GeForce RTX 3090 GPU.

4.3 Results

In our study, we presented the prediction accuracy of affiliation, along with the corresponding point accuracy (PA) and range prediction (PW) scores. The results indicate that our method performs well across multiple datasets. Although it did not achieve the best results on the UCR dataset, it maintained a strong competitive edge relative to other methods. It is noteworthy that RCF and LSTM exhibited excellent performance on the UCR dataset, but their accuracy significantly declined on the AIOps dataset. This phenomenon can be attributed to the fact that the UCR dataset typically contains only a single anomaly segment and lacks anomalous samples in the training set. In contrast, the AIOps dataset features multiple anomaly segments, and the training set includes some anomalous characteristics, leading to higher false negative rates for methods that lack robustness.

To enhance the robustness of the model, the AOC method employs a soft-boundary strategy, which effectively improves the model's adaptability to anomalies. In contrast, our approach integrates the RPCA method to successfully filter out a significant portion of anomalous features in the training set. This approach also yielded favorable results on the multivariate time series dataset SMAP, further validating the broad applicability of our method.

As shown in Table 2, in comparisons with various baseline models, we draw the following conclusions: RCF, as a shallow model, demonstrated outstanding performance on the UCR dataset, even surpassing some deep learning models. Meanwhile, two-stage anomaly detection methods, including SVDD and TS-TCC, did not achieve ideal results in time series anomaly detection, revealing the limitations of staged approaches for time series data and thereby constraining the performance of deep models. Additionally, our proposed ROC method performed well across all three datasets, confirming the effectiveness of the ensemble approach and its robustness against contaminated training sets.

Table 2: Results summary

Datasets	Metric	SVM	RCF	LSTM	DAGMM	SVDD	TS-TCC	AOC	ROC	NoRPCA	NoOC
AIOps	Precision	45.8	52.6	52.2	44.7	47.3	50.5	90.3	**96.2**	94.3	92.2
	Recall	17.5	25.7	25.3	30.4	32.1	23.4	**38.6**	36.7	35.1	34.8
	F1-score	25.4	34.5	34.1	36.2	38.2	31.9	**54.0**	53.1	51.2	50.5
	PA	53.4	53.2	76.1	14.2	14.3	17.5	80.1	**86.3**	81.6	62.3
	PW	8.7	16.6	6.4	5.8	6.4	13.4	45.5	**47.2**	45.3	45.1

(Continued)

Table 2 (continued)

Datasets	Metric	SVM	RCF	LSTM	DAGMM	SVDD	TS-TCC	AOC	ROC	NoRPCA	NoOC
UCR	Precision	47.6	59.1	**67.8**	51.2	37.2	44.3	61.7	65.3	50.3	51.2
	Recall	**84.0**	57.7	66.0	96.7	37.1	44.3	61.5	63.3	60.3	61.8
	F1-score	60.3	58.4	**66.9**	66.9	37.2	44.3	61.6	64.1	54.8	56.0
	PA	10.1	**98.2**	97.8	11.3	78.2	62.3	62.6	94.9	63.2	94.8
	PW	2.02	35.3	**43.2**	6.6	8.3	14.2	15.3	22.6	17.6	16.8
SMAP	Precision	43.3	42.2	84.3	40.6	51.9	45.4	91.3	**95.6**	93.7	90.6
	Recall	34.2	**52.3**	24.3	15.1	46.9	17.2	36.3	41.7	38.6	39.1
	F1-score	38.2	46.7	37.7	22.0	49.3	24.9	51.9	**58.1**	54.7	54.6
	PA	97.1	90.2	**98.5**	86.0	86.6	94.4	86.1	90.2	88.0	87.6
	PW	14.1	7.4	**49.0**	7.0	13.1	12.2	37.2	40.9	37.6	35.3

On some datasets, although our methods are not the best, they do not fall behind much. The reason why our method does not achieve the best performance on the UCR dataset may lie in the fact that many time series in the UCR dataset exhibit strong contextual dependencies [32,33]. For instance, in motion sensor data, transitions between different actions, or in weather data, long-term trends and seasonal patterns play a significant role. Models like LSTM, which excel at handling strongly time-dependent sequential data, can effectively capture critical patterns through learning temporal transitions between states. This capability allows LSTM to achieve higher accuracy in tasks such as behavior classification and anomaly detection. As a result, methods like LSTM are more suitable for datasets with strong temporal dependencies, such as UCR.

However, for datasets like AIOps, which encompass rich operational data and diverse task scenarios, our method demonstrates superior performance. This is due to its ability to handle large-scale data with highly diverse anomaly samples and to tackle more complex tasks. In such cases, our method significantly outperforms LSTM and other approaches that rely solely on temporal dependencies.

Finally, the results from NoOC and NoRPCA indicate that combining multiple normality assumptions with anomaly filtering models significantly enhances anomaly detection (AD) performance, this further validates the efficacy and importance of the different elements within our model.

4.4 Hyper-Parameter Analysis

In this section, we perform a hyperparameter analysis on the AIOps dataset, with a specific focus on examining two key parameters: λ_1 and λ_2 in the equations. Fig. 3a illustrates the results of varying λ_1 for RPCA, showing that the model performs best when $\lambda_1 = 0.01$. We hypothesize that the underlying reason for the observed performance may be that when the score from the one-class method constitutes a larger proportion of the anomaly score and exceeds the threshold value of 0.01, the model's performance tends to approximate that of shallow methods such as SVM. Fig. 3b demonstrates the impact of λ_2 on overall performance, with the model achieving optimal results when $\lambda_2 = 0.5$, identifying the best threshold for filtering anomaly features from the training set. The y-axis in the figure represents PA and PW precision metric.

We also conducted detailed experiments to investigate the reasons behind the performance decline associated with varying λ_2. As a parameter that adjusts the sparsity in S, λ_2 plays a crucial role in our analysis. Specifically, a smaller λ_2 encourages a large amount of data to be isolated as noise or anomalies in S, which minimizes the reconstruction error of the autoencoder; however, this can severely distort the original time series, resulting in inadequate anomaly detection due to low anomaly scores. Conversely, a larger λ_2 prevents

data from being classified as noise or anomalies, leading to increased reconstruction errors. As shown in the Fig. 4, when λ_2 is too large, the reconstruction error rises, and since only a few anomalies are isolated, the results resemble those prior to RPCA, thus losing the filtering effect and causing a decline in performance. The y-axis in the figure represents the model's anomaly scores across different batches.

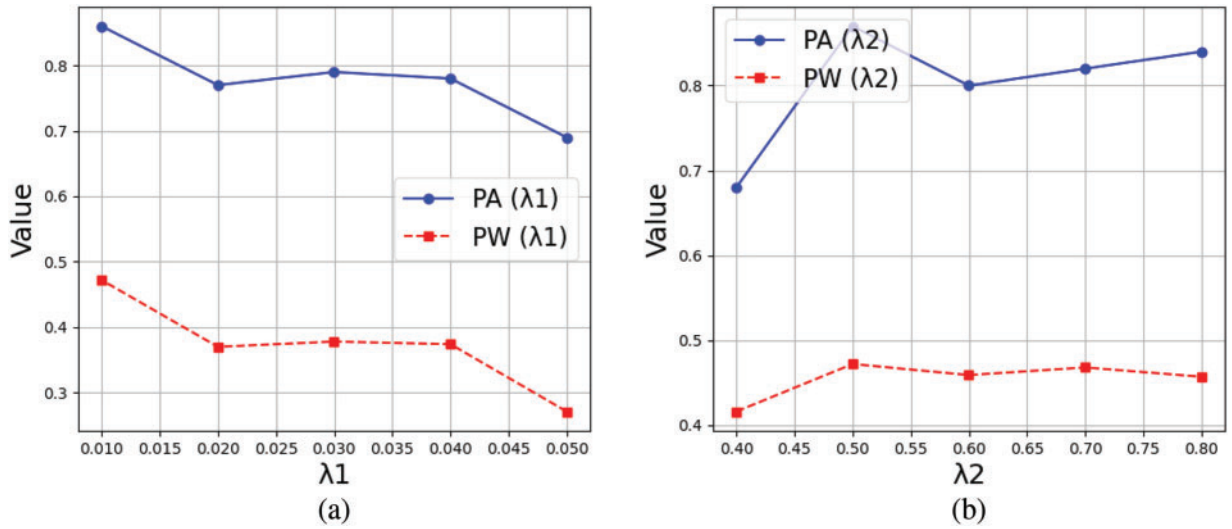

Figure 3: We conducted a hyperparameter analysis on AIOps for λ_1 (**a**) and λ_2 (**b**), focusing on the impact of anomaly filtering on the model's precision

Figure 4: The figure illustrates how the anomaly scores change with variations in λ_2, where the green curve represents the scenario in which λ_2 is at its optimal value. This configuration more accurately reflects the anomaly scores of the time series, providing a clearer indication of the actual anomalies present

4.5 Optimization Algorithm Analysis

We employ the **gradient descent method** as the core optimization algorithm to minimize the objective function. Gradient descent iteratively updates the parameters by computing the gradient of the loss function with respect to the model's parameters, ensuring a systematic approach to minimizing loss.

In addition, we incorporate a **dynamically adjusted learning rate** during the optimization process. The dynamic adjustment of the learning rate allows the algorithm to take larger steps when far from the

optimal solution to accelerate convergence, while automatically reducing the step size as it approaches the optimal solution. This mechanism helps to avoid overshooting the minimum and improves stability near the global optimum. More importantly, the dynamically adjusted learning rate mitigates the risk of the algorithm getting trapped in local minima, a common issue in non-convex optimization problems.

By comparing the effects of dynamic and fixed learning rates, Fig. 5 provides a visual representation of how the convergence behavior differs under different learning rate strategies. The dynamic learning rate strategy demonstrates faster convergence and better adaptability to the optimization landscape, particularly in complex scenarios.

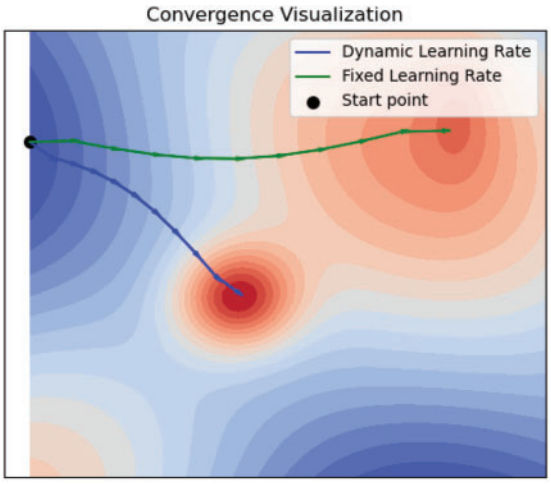

Figure 5: Figure demonstrates the convergence process of the optimization algorithm

In addition, we visualized the actual convergence process of the model, demonstrating the step-by-step reduction in the loss function values during optimization. Fig. 6 not only provides an intuitive comparison between the performance of dynamic learning rate adjustment and fixed learning rate strategies but also strongly supports the effectiveness of the dynamic learning rate. Specifically, the dynamic learning rate facilitates a faster reduction in loss values and exhibits greater stability as it approaches the global optimum. This indicates that the dynamic learning rate has significant advantages in optimizing non-convex problems and handling complex objective functions. Furthermore, it validates the applicability of this approach in addressing challenging tasks.

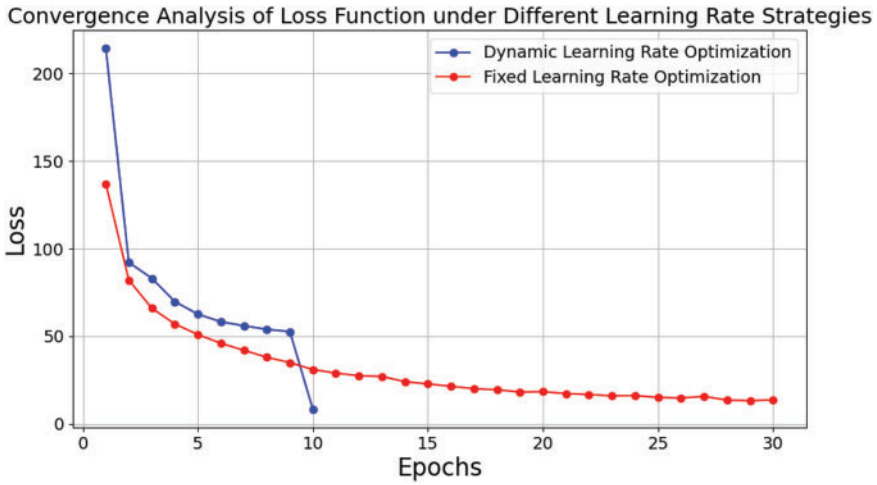

Figure 6: The optimization process with dynamic learning rate and fixed learning rate as a function of epochs

5 Conclusion

This paper introduces a robust time series anomaly detection method, **ROC**, which is grounded in multiple hypotheses and eliminates the need for pre-training. The proposed method projects the hidden representation layer of the autoencoder and integrates the objectives of both the autoencoder and one-class (OC) methods. By filtering out anomalous segments of the input time series, **ROC** avoids the contamination of the compression layer by anomalous features during training and resolves potential inconsistencies between the two hypotheses. This approach effectively captures normal patterns from multiple perspectives, allowing the model to learn a more comprehensive representation of typical time series data. As a result, the method demonstrates an enhanced ability to detect diverse types of anomalies. Experimental evaluations on three real-world datasets validate the superior performance of the proposed approach.

In future work, we plan to further enhance the method's robustness against adversarial attacks in time series anomaly detection. Drawing inspiration from state-of-the-art techniques, we aim to explore feature learning from various forms of time series representations, such as residuals and frequency domains. Additionally, we intend to combine these advanced feature extraction techniques with our robust approach to filter anomalous features from multiple perspectives, ultimately improving the model's effectiveness and adaptability in complex scenarios.

Acknowledgement: The authors appreciate the reviewers and editors for their valuable feedback on this work. We also acknowledge the providers of datasets, including AIOps and UCR.

Funding Statement: This research was supported by the National Natural Science Foundation (62202118), Guizhou Province Major Project (Qiankehe Major Project [2024]014), Science and Scientific and Technological Research Projects from Guizhou Education Department (Qianiao ji [2023]003), Hundred-level Innovative Talent Project of Guizhou Provincial Science and Technology Department (Qiankehe Platform Talent-GCC[2023]018) and Guizhou Province Major Project (Qiankehe Major Project [2024]003), Foundation of Chongqing Key Laboratory of Public Big Data Security Technology (CQKL-QJ202300001).

Author Contributions: The authors confirm contribution to the paper as follows: method proposal and implementation, experimental proof, and manuscript writing: Zhengdao Yang; experimental setting and grant support: Yuling Chen, Xuewei Wang; manuscript revision: Hui Dou, Haiwei Sang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used to support the findings of this study are available from the corresponding author upon request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Nomenclature

AE	Autoencoder
AD	Anomaly Detection
ADMM	Alternating Direction Method of Multipliers
AIOps	Artificial Intelligence for IT (Information Technology) Operations Performance Score
AOC	deep Autoencoding One Class
BACKPROP	Backpropagation
DDoS	Distributed Denial of Service
DAGMM	Deep Autoencoding Gaussian Mixture Model
EMA	Exponential Moving Average
ECG	Electrocardiogram
EEG	Electroencephalogram
GAN	Generative Adversarial Network
KPI	Key Performance Indicator
LSTM	Long Short-Term Memory
MP	Matrix Profile
OC	One-Class Classification
OS	Outlier Score
PCA	Principal Component Analysis
PA	Point-adjusted metrics
PW	Point-wise metrics
RCF	Random Cut Forest
RPCA	Robust Principal Component Analysis
ROC	Robust One-Class Classification Detection
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SSA	Singular Spectrum Analysis
STL	Seasonal and Trend decomposition using Loess
SVDD	Support Vector Data Description
TSAD	Time Series Anomaly Detection
TS-TCC	Time-Series representation learning via Temporal and Contextual Contrasting
UCR	University of California, Riverside Time Series Anomaly Archive

References

1. Grubbs FE. Procedures for detecting outlying observations in samples. Technometrics. 1969;11(1):1–21. doi:10.1080/00401706.1969.10490657.
2. Pang G, Shen C, Cao L, Hengel AVD. Deep learning for anomaly detection: a review. ACM Comput Surv. 2021;54(2):1–38. doi:10.1145/3439950.
3. Sun Y, Chen Y, Wu P, Wang X, Wang Q. DRL: dynamic rebalance learning for adversarial robustness of UAV with long-tailed distribution. Comput Commun. 2023;205(6):14–23. doi:10.1016/j.comcom.2023.04.002.
4. Yang H, Wang X, Chen Y, Dou H, Zhang Y. RPU-PVB: robust object detection based on a unified metric perspective with bilinear interpolation. J Cloud Comput. 2023;12(1):169. doi:10.1186/s13677-023-00534-3.

5. Yang X, Chen Y, Qian X, Li T, Lv X. BCEAD: a blockchain-empowered ensemble anomaly detection for wireless sensor network via isolation forest. Secur Commun Netw. 2021;2021(1):9430132. doi:10.1155/2021/9430132.
6. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–7. doi:10.1126/science.1127647.
7. Tishby N, Pereira FC, Bialek W. The information bottleneck method. arXiv:physics/0004057. 2000. doi:10.48550/arXiv.physics/0004057.
8. Lee B, Kim S, Maqsood M, Moon J, Rho S. Advancing autoencoder architectures for enhanced anomaly detection in multivariate industrial time series. Comput Mater Contin. 2024;81(1):1275–300. doi:10.32604/cmc.2024.054826.
9. Bar S, Prasad P, Sayeed MS. Enhancing internet of things intrusion detection using artificial intelligence. Comput, Mater Contin. 2024;81(1):1–23. doi:10.32604/cmc.2024.053861.
10. Long G, Zhang Z. PUNet: a semi-supervised anomaly detection model for network anomaly detection based on positive unlabeled data. Comput Mater Contin. 2024;81(1):327–43. doi:10.32604/cmc.2024.054558.
11. Blázquez-García A, Conde A, Mori U, Lozano JA. A review on outlier/anomaly detection in time series data. ACM Comput Surv. 2021;54(3):1–33. doi:10.1145/3444690.
12. Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, et al. Deep one-class classification. In: International Conference on Machine Learning; 2018; PMLR. p. 4393–402.
13. Sohn K, Li CL, Yoon J, Jin M, Pfister T. Learning and evaluating representations for deep one-class classification. arXiv:2011.02578. 2020. doi:10.48550/arXiv.2011.02578.
14. Guo R, Liu H, Liu D. When deep learning-based soft sensors encounter reliability challenges: a practical knowledge-guided adversarial attack and its defense. IEEE Trans Ind Inform. 2023;20(2):2702–14. doi:10.1109/TII.2023.3297663.
15. Guo R, Chen Q, Liu H, Wang W. Adversarial robustness enhancement for deep learning-based soft sensors: an adversarial training strategy using historical gradients and domain adaptation. Sensors. 2024;24(12):3909. doi:10.3390/s24123909.
16. Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging; 2017; Springer. p. 146–57.
17. Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv:1607.00148. 2016. doi:10.48550/arXiv.1607.00148.
18. Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations; 2018.
19. Qiu C, Pfrommer T, Kloft M, Mandt S, Rudolph M. Neural transformation learning for deep anomaly detection beyond images. In: International Conference on Machine Learning; 2021; PMLR. p. 8703–14.
20. Candès EJ, Li X, Ma Y, Wright J. Robust principal component analysis? J ACM. 2011;58(3):1–37. doi:10.1145/1970392.197039.
21. Donoho DL. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. Commun Pure Appl Math: A J Issued Courant Institute Math Sci. 2006;59(6):797–829. doi:10.1002/cpa.20132.
22. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends® Mach Learn. 2011;3(1):1–122. doi:10.1561/2200000016.
23. Parikh N, Boyd S, et al. Proximal algorithms. Found Trends® Optim. 2014;1(3):127–239. doi:10.1561/2400000003.
24. Dau HA, Bagnall A, Kamgar K, Yeh CCM, Zhu Y, Gharghabi S, et al. The UCR time series archive. IEEE/CAA J Autom Sin. 2019;6(6):1293–305. doi:10.1109/JAS.2019.1911747.
25. Schölkopf B, Williamson RC, Smola A, Shawe-Taylor J, Platt J. Support vector method for novelty detection. In: Solla S, Leen T, Müller K, editors. Advances in neural information processing systems. MIT Press; 1999.
26. Guha S, Mishra N, Roy G, Schrijvers O. Robust random cut forest based anomaly detection on streams. In: International Conference on Machine Learning; 2016; PMLR. p. 2712–21.
27. Eldele E, Ragab M, Chen Z, Wu M, Kwoh CK, Li X, et al. Time-series representation learning via temporal and contextual contrasting. arXiv:2106.14112. 2021. doi:10.24963/ijcai.2021.

28. Mou X, Wang R, Wang T, Sun J, Li B, Wo T, et al. Deep autoencoding one-class time series anomaly detection. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2023, IEEE. p. 1–5.
29. Huet A, Navarro JM, Rossi D. Local evaluation of time series anomaly detection algorithms. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2022. p. 635–45.
30. Schmidl S, Wenig P, Papenbrock T. Anomaly detection in time series: a comprehensive evaluation. Proc VLDB Endow. 2022;15(9):1779–97. doi:10.14778/3538598.3538602.
31. Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019. p. 2828–37.
32. Wu R, Keogh EJ. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. IEEE Trans Knowl Data Eng. 2021;35(3):2421–9. doi:10.1109/TKDE.2021.3112126.
33. Lai KH, Zha D, Xu J, Zhao Y, Wang G, Hu X. Revisiting time series outlier detection: definitions and benchmarks. In: Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1); 2021.