Knowledge-Based Systems

Boundary-Aware Adversarial Ensemble Learning for Multivariate Time Series Anomaly Detection

--Manuscript Draft--

Manuscript Number:	KNOSYS-D-24-13045
Article Type:	Full Length Article
Keywords:	safety-critical system; multivariate time series; anomaly detection; uncertainty estimation
Abstract:	In safety-critical systems like aircraft, manufacturing, and energy, anomalies often signal early issues, and failing to detect them promptly can lead to severe losses of life and property. In these scenarios, anomaly detection faces challenges including the scarcity of anomalous data and the complexity and diversity of fault modes. To address these challenges, we proposes a novel boundary-aware anomaly detection method in which training data are augmented with boundary samples near the normal data. In contrast with other data augmentation techniques, the boundary samples are beneficial to anomaly detection in that it provide a natural and convenient computational mechanism to incorporate prior knowledge about the abnormality while maintaining their diversity. For this purpose, we use an ensemble-based model to infer the uncertainty about the degree of abnormality of the candidate samples, and only accept those with low uncertainty, while those samples themselves are produced by an end-to-end trained deep generative model. Extensive experiments demonstrate that our method outperforms 13 existing methods across 5 categories, and achieves statistically significant improvements. The experiments results also validate the critical role of uncertainty in boundary sample generation and offer a new perspective for multivariate time-series anomaly detection in safety-critical systems.

Boundary-Aware Adversarial Ensemble Learning for Multivariate Time Series Anomaly Detection

Pengcheng He[a], Xiaoyang Tan[a,*], Yuehua Cheng[b]

[a] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing, 211106, Jiangsu, China*
[b] *College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, Jiangsu, China*

Abstract

In safety-critical systems like aircraft, manufacturing, and energy, anomalies often signal early issues, and failing to detect them promptly can lead to severe losses of life and property. In these scenarios, anomaly detection faces challenges including the scarcity of anomalous data and the complexity and diversity of fault modes. To address these challenges, we proposes a novel boundary-aware anomaly detection method in which training data are augmented with boundary samples near the normal data. In contrast with other data augmentation techniques, the boundary samples are beneficial to anomaly detection in that it provide a natural and convenient computational mechanism to incorporate prior knowledge about the abnormality while maintaining their diversity. For this purpose, we use an ensemble-based model to infer the uncertainty about the degree of abnormality of the candidate samples, and only accept those with low uncertainty, while those samples themselves are produced by an end-to-end trained deep generative model. Extensive experiments demonstrate that our method outperforms 13 existing methods across 5 categories, and achieves statistically significant improvements. The experiments results also validate the critical role of uncertainty in boundary sample generation and offer a new perspective for multivariate time-series anomaly detection in safety-critical systems.

*Corresponding author.
Email addresses: yukina233@nuaa.edu.cn (Pengcheng He), x.tan@nuaa.edu.cn (Xiaoyang Tan), chengyuehua@nuaa.edu.cn (Yuehua Cheng)

Keywords: safety-critical system, multivariate time series, anomaly detection, uncertainty estimation

1. Introduction

Anomalies are generally defined as observations that deviate from typical behavior patterns. In safety-critical systems like aircraft, industrial manufacturing and energy systems, anomalies are often early indicators of potential system failures. Failing to detect anomalies promptly can result in severe outcomes, including system breakdowns, accidents, and significant loss of life or property. The growing complexity of modern systems and diverse operating environments have heightened uncertainties in system operations, which makes anomaly detection more critical than ever. In such scenarios, anomalies often manifest as abnormal changes in the relationships among multiple variables, which makes multivariate time series anomaly detection a key tool for identifying these patterns[1, 2, 3].

However, applications of multivariate time series anomaly detection in safety-critical contexts involves several challenges[4, 5]. For example, in aircraft fault diagnosis, systems are usually in the normal state, which leads to a scarcity of anomalous data. Additionally, the complex interactions and diverse fault modes in such systems make anomalies highly unpredictable, which greatly increases the need for anomalous samples. Safety-critical applications also impose strict requirements, as anomalies may lead to high-risk events—faults in aircraft systems can threaten lives, industrial failures may halt production, and energy system anomalies could cause blackouts. Therefore, anomaly detection in these scenarios must prioritize reliability and conservativeness, to ensure the accurate identification of abnormal behaviors under various circumstances.

Data-driven approaches, particularly those utilize deep learning, have become a primary research focus for anomaly detection in safety-critical systems due to their flexibility and adaptability[5, 6]. However, deep learning methods typically require large volumes of historical data, which poses significant challenges in fields like aircraft fault diagnosis, where anomalous data are scarce and anomaly patterns are highly complex. To address these challenges, many existing approaches focus exclusively on modeling normal behavior patterns. These methods identify anomalies by measuring deviations from normal patterns, and capture the intrinsic structure and behavior of

normal states[7]. By avoiding reliance on anomalous samples or assumptions about specific anomaly distributions, these approaches partially mitigate the issues arising from the lack of anomalous data and the diversity of anomaly patterns. However, studies have shown that deep anomaly detection methods trained solely on normal data often exhibit overconfidence, frequently overestimating the normality of anomalous samples. This results in unreliable detection and an increased likelihood of missing anomalies[8, 9, 10]. This limitation arises primarily from the absence of supervision from anomalous data, which prevents the model from learning sufficiently discriminative features and patterns. As a result, while these models are effective at assigning high probabilities to normal samples, they struggle to consistently assign low probabilities to truly anomalous samples.

To tackle these challenges, this paper proposes the Boundary-aware Adversarial Ensemble Model (BAEM), a novel framework for multivariate time series anomaly detection that relies only on normal data, without requiring any anomalous samples. As a small number of anomalous samples cannot sufficiently represent the region outside the normal data distribution, we incorporate prior knowledge to generate boundary samples near the edges of the distribution of the normal data, and then use these samples to calibrate the detection model. Specifically, we use an ensemble model to estimate the uncertainty of generated samples, ensuring that boundary samples exhibit low uncertainty and align with their intended characteristics.

Extensive experiments were conducted on multiple real-world and simulated datasets. Compared against 13 approaches across 5 categories, the results show that the BAEM method consistently outperforms existing methods in terms of AUROC and AUPR metrics across various random seeds. Furthermore, statistical tests confirm that the performance improvements achieved by BAEM are statistically significant. In summary, the main contributions of this paper are as follows:

- We propose a novel anomaly detection framework for multivariate time series that addresses the challenges posed by the scarcity of anomalous data and the diversity of anomaly patterns. The method relies exclusively on known normal samples and leverages prior knowledge to generate samples that approximate the edge of the normal data. These boundary samples are then used to provide additional supervision and improve the model's ability to distinguish between normal and anomalous states. To guide the procedure of boundary sample generation, we

introduce an ensemble model that evaluates the uncertainty of generated samples, ensuring that they possess boundary characteristics and enhance the model's generalization.

- We show that in data augmentation for anomaly detection, uncertainty plays a critical role in controlling the quality of generated samples. In particular, we demonstrate that low-uncertainty samples significantly boost the detector's performance by providing reliable supervisory signals. To the best of our knowledge, this observation has not been reported in the anomaly detection literature, offering a new perspective on improving anomaly detection methods.

2. Related works

2.1. Multivariate Time-series Anomaly Detection

In recent years, the growing complexity of modern systems and devices has brought significant attention to the application of deep learning in multivariate time series anomaly detection. These methods have shown exceptional performance in various fields, such as fault detection in aircraft and industrial systems[11, 12, 13], anomaly detection in medical applications[14, 15], and network intrusion detection[16, 17, 18].

In deep anomaly detection for multivariate time series, the limited availability of anomalous data and the diversity of anomaly patterns have driven most methods to adopt unsupervised or semi-supervised approaches that rely exclusively on normal data. Many studies focus on learning normal behavior patterns from training data and identifying anomalies as deviations from these patterns.

Among reconstruction-based approaches, autoencoders (AEs) are among the most commonly used methods [19, 20]. These methods are designed to learn and reconstruct representative features of normal patterns. Since anomalies are often characterized by non-representative features, are challenging for autoencoders to reconstruct. This enables anomaly detection using the residual distance between original and reconstructed features. To capture temporal correlations within variables, some researchers extract statistical features from overlapping sliding windows before using autoencoders[21].

Probability-based methods, such as Variational Autoencoders (VAE) [11], Generative Adversarial Networks (GAN) [22], and Empirical Cumulative Distribution-based Outlier Detection (ECOD) [23], can model normal data

distributions probabilistically and identify low-probability points as anomalies. Since generative models can not only model probability distributions but also provide data reconstruction, some studies utilize their reconstruction errors to detect anomalies [24, 25].

One-class classification methods are also widely used for time series anomaly detection. These methods do not estimate density directly, instead, they learn a decision boundary aligned with the expected density level of normal data or optimize for low error on unseen data. Common approaches include Deep SVDD and OC-SVM[26, 27].

However, the deep anomaly detection methods mentioned above often suffer from overconfidence, which can result in overestimating the normality of anomalous samples. This issue undermines the reliability of detection results and increases the risk of missing anomalies [8, 9, 10].

2.2. Anomaly Detection with Augmented Data

To obtain additional information of anomalous patterns, some studies construct reference anomaly datasets as augmented data for calibration. Methods using external reference datasets as anomalous samples have been studied in the computer vision domain[28, 29, 30]. For example, the Outlier Exposure (OE) approach avoids assumptions about anomaly distributions and directly utilizes existing image and text databases as auxiliary anomaly data to improve deep anomaly detection[29, 30]. However, the diversity of data structures across tasks makes these methods difficult to generalize to multivariate time series anomaly detection.

Conversely, some studies aim to augment anomaly datasets by utilizing known internal data and employing generative models, such as GANs, to generate boundary samples near the edge of normal data distribution. These methods use anomaly scores as thresholds to control the level of anomaly. For instance, OCAN[22] uses a GAN discriminator to approximate the probability density function of normal samples, constraining its output to ensure generated samples lie in low-density regions. FenceGAN[31] controls anomaly scores of generated samples and ensures diversity by maximizing their distance from centers in the original space. Although these studies employ adversarial training to generate boundary samples from normal data distributions for the calibration of the anomaly detection model [32, 33, 31, 22], they overlook the inherent inaccuracies in the estimations of the detection model during the boundary sample generation process. This oversight compromises the effectiveness of the calibration.

2.3. Ensemble Learning

Ensemble learning is a machine learning technique that employs multiple learners to solve the same problem, enhancing prediction performance and generalization by combining their outputs. The core concept of ensemble learning is to combine multiple weak learners into a strong learner, achieving greater predictive accuracy than individual models[34].

Common ensemble methods include Bagging [35], which involves randomly sampling subsets of training data, training submodels in parallel, and aggregating their predictions. AdaBoost [36] trains models sequentially, iteratively adjusts sample weights to emphasize misclassified instances, and enhances the performance of subsequent submodels. XGBoost [37] employs gradient descent to minimize a loss function, trains models sequentially to address residual errors, and iteratively refines predictions.

Deep ensemble learning has demonstrated outstanding performance and reliability in uncertainty estimation tasks [38]. Inspired by this, we adopt a deep ensemble strategy to evaluate uncertainty in our approach.

3. Proposed method

This chapter introduces our anomaly detection framework for multivariate time series in safety-critical systems. We begin by formally presenting the problem statement and associated challenges, followed by a detailed explanation of the proposed method.

3.1. Problem Statement

Anomaly detection in multivariate time series is a critical task in various domains, including safety-critical systems such as aircraft, industrial manufacturing, and energy systems. The goal is to identify abnormal data points that deviate from expected patterns, as these anomalies often indicate potential system failures or malfunctions. However, this task is challenging due to the scarcity of labeled anomalous data, the diversity of anomaly patterns, and the complex dependencies among variables over time.

Formally, let $X = \{x_1, x_2, \cdots, x_T\}$ represent a multivariate time series, where each time step $t = 1, \cdots, T$ corresponds to an observation vector $x_t \in \mathbb{R}^d$, with $d > 1$ denoting the number of variables. The corresponding labels are denoted as $Y = \{y_1, y_2, \cdots, y_T\}$, where $y_t \in \{0, 1\}$. The label $y_t = 0$ indicates that the data point is normal, while $y_t = 1$ indicates an anomaly.

In the unsupervised setting, the training dataset $D_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^{M}$ is assumed to contain only normal time series, meaning that for every $(X_i, Y_i) \in D_{\text{train}}$, $y_t^i = 0, \forall t$. The objective of anomaly detection is to predict the labels Y_{new} for a new time series X_{new}, identifying which time steps are anomalous.

Since only normal data is available during training, the anomaly detection task involves learning a model of normal behavior and identifying anomalies based on deviations from the learned patterns. To quantify these deviations, an anomaly score function $s(x)$ is defined, where higher scores indicate a stronger likelihood of a data point x being anomalous. Based on a threshold τ, anomaly scores are converted into binary labels as follows:

$$\hat{y} = \mathbb{1}\big(s(x) \geq \tau\big) \tag{1}$$

where $\hat{y}$ is the predicted label for x, $\mathbb{1}(\cdot)$ is the indicator function, and τ is the threshold. So the problem lies in how to learn an accurate anomaly scoring function $s(x)$ that generalizes well to unseen anomalies, despite only being trained on normal data.

This paper focuses on improving the accuracy of the anomaly scoring function by leveraging generated boundary samples. These samples, positioned near the boundary of the normal data distribution, provide supervisory signals that enhance the model's ability to distinguish between normal and anomalous data. By addressing the challenges of data scarcity and complex anomaly patterns, our approach aims to improve anomaly detection performance in safety-critical systems.

3.2. Methodology

3.2.1. Time-Domain Features Extraction

To preprocess multivariate time series data, we segment raw signals into fixed-length subsequences using a sliding window approach as other works in time series analysis[39, 40]. For a given window length L, the time series X is divided into overlapping windows. Let $W_L(X)$ represent the set of all windows generated from X. Each window $w \in W_L(X)$ is assigned a label y, where $y = 1$ if any time step within the window is anomalous, and $y = 0$ otherwise.

By this segmentation method, the original dataset $D_{\text{train}} = (X_i, Y_i)_{i=1}^{M}$ is transformed into a windowed dataset $D_W = \{(w, y) \mid w \in W_L(X), X \in D_{\text{train}}\}$. This new dataset consists of pairs of windows and their corresponding labels, which allows for more localized analysis of time-series behavior.

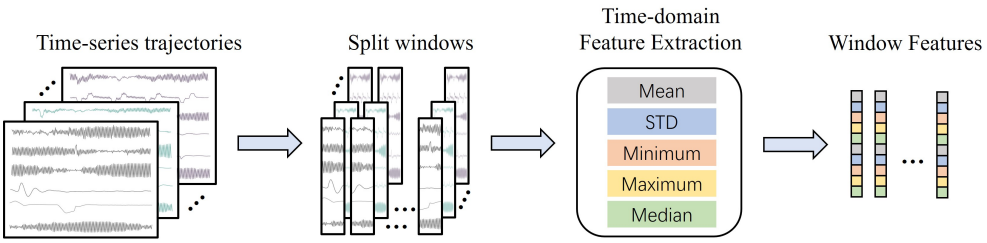


Figure 1: The time-domain features extraction process of multivariate time-series data. First split each trajectory using sliding windows, then extract time-domain features from each window.

Next, time-domain features are extracted from each window to capture statistical patterns in the raw signals. Specifically, we compute five statistical features: mean, standard deviation, minimum, maximum, and median. These features have proven effective in capturing temporal information, as demonstrated by their strong performance on our proprietary aircraft dataset. A visual representation of this process is shown in Figure 1. The same preprocessing method was applied to all datasets used in the experimental section.

3.2.2. Uncertainty-Aware Boundary Data Generation

In real scenarios, deep anomaly detectors are typically trained using only normal samples. This limitation causes anomalous samples to become out-of-distribution (OOD) for the model, which leads to biased detection results and inaccuracies in the anomaly detection process. In this context, many existing studies generate boundary samples by using the anomaly scores provided by detection models to capture the boundary characteristics of samples. As outlined in Section 2.2, high anomaly scores are commonly used as the main constraint in boundary data generation. However, current methods fail to ensure that the generated samples with high anomaly scores accurately represent the boundary region of the normal data distribution. They overlook the inherent inaccuracies of anomaly detectors, which can result in boundary samples that do not truly have boundary characteristics. This behavior leads to a risk of incorrect boundary calibration. A visualization of boundary samples generated using only anomaly score constraints is shown in Figure 2(left). The results indicate that the generator tends to produce samples within the normal data distribution rather than at the boundary region, undermining their utility in calibration. In contrast, our proposed method as

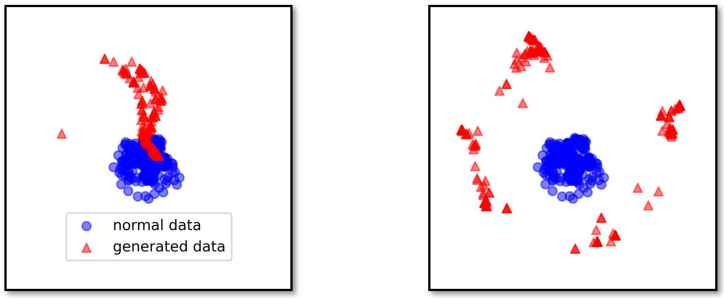


Figure 2: Comparison of boundary samples generated by existing methods and our proposed approach. The left figure illustrates samples generated using the existing criterion (anomaly score), while the right figure presents samples generated using our method with the added uncertainty constraint. Further details can be found in Section 4.4.

illustrated in Figure 2(right), successfully generates boundary samples that are distinguishable from normal data and better represent the desired boundary region.

In particular, in this work we proposed to guide the generating procedure based on the confidence of the detection model in its predictions, i.e. uncertainty estimation. By measuring the uncertainty in the model's predictions, we can mitigate the influence of inaccurate estimations on the generation process. This ensures that the generated samples have better quality, making them more reliable for calibrating anomaly detection models. Consequently, this approach improves the model's generalization and robustness. The overall framework of the proposed method is depicted in Figure 3.

We implement the above idea within the adversarial training framework of GAN. In GAN, a generator G produces samples by transforming random vectors $z \sim P_{pri}(z)$ drawn from a prior distribution, while a discriminator D distinguishes between real and generated samples. The classic GAN optimization objective is defined as:

$$\min_{G} \max_{D} \mathbb{E}_{x \sim P_{in}(x)} \big[\log D(x) \big] + \mathbb{E}_{z \sim P_{pri}(z)} \big[\log(1 - D(G(z))) \big] \tag{2}$$

where $P_{in}(x)$ denotes the normal data distribution. However, unlike traditional GANs that aim to mimic the normal data distribution, our objective is to generate samples located near the decision boundary of normal data.

To generate effective boundary samples, the generator's optimization objective is modified to satisfy the following two conditions:

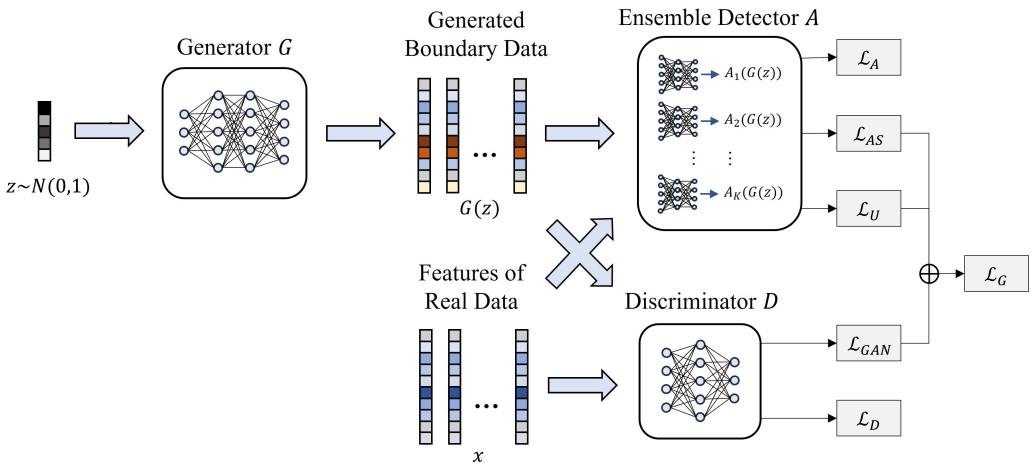


Figure 3: Framework of the proposed adversarial training process. The generator G is trained by loss function $\mathcal{L}_G$ (equation 6), including $\mathcal{L}_{AS}$ (equation 3) and $\mathcal{L}_U$ (equation 4) from the ensemble detector A and $\mathcal{L}_{GAN}$ (equation 5) from discriminator D. Each submodel A_k in the ensemble detector A is trained by loss function $\mathcal{L}$ (equation 7) using real data and the generated boundary data. The discriminator D is trained by $\mathcal{L}_D$(equation 8) as in the original GAN.

- High anomaly scores: The generated samples must have high anomaly scores, which indicates proximity to the decision boundary.
- Low uncertainty: The predictions of ensemble submodels for the generated samples must be consistent, i.e., have low standard deviation.

To achieve this, we introduce two key loss terms for the generator: the anomaly score loss ($\mathcal{L}_{AS}$) and the uncertainty loss ($\mathcal{L}_U$), defined as follows:

$$\mathcal{L}_{AS} = -\mathbb{E}_{z \sim P_{pri}(z)}\big[A(G(z))\big] \tag{3}$$

$$\mathcal{L}_U = \mathbb{E}_{z \sim P_{pri}(z)}\big[u(A(G(z)))\big] \tag{4}$$

In these equations, A represents an anomaly detection model. The term $A(G(z))$ denotes the anomaly score assigned to the generated sample $G(z)$, while $u(A(G(z)))$ quantifies the uncertainty of the anomaly score predictions. Details on the computation of uncertainty are provided in the next section. The anomaly score loss $\mathcal{L}_{AS}$ ensures that the generated samples have high anomaly scores, corresponding to their proximity to the decision boundary.

The uncertainty loss $\mathcal{L}_U$ minimizes the standard deviation of predictions, ensuring low uncertainty of boundary sample.

To prevent the generated samples from straying too far from the normal data distribution, we retain the original GAN discriminator D, which distinguishes whether a sample resembles the normal data. The corresponding GAN loss $\mathcal{L}_{GAN}$ is defined as:

$$\mathcal{L}_{GAN} = \mathbb{E}_{z \sim P_{pri}(z)} \Big[\log(1 - D(G(z))) \Big] \tag{5}$$

The overall loss for the generator $\mathcal{L}_G$ combines these components:

$$\mathcal{L}_G = \mathcal{L}_{GAN} + \alpha \cdot \mathcal{L}_{AS} + \beta \cdot \mathcal{L}_U \tag{6}$$

On the other side, the anomaly detector A is trained adversarially with the generator. The corresponding loss function $\mathcal{L}_A$ is:

$$\mathcal{L}_A = \mathbb{E}_{x \sim P_{in}(x)} \big[A(x) \big] + \eta \cdot \mathbb{E}_{z \sim P_{pri}(z)} \big[A(G(z)) \big]^{-1} \tag{7}$$

The first term minimizes the anomaly scores for normal samples, while the second term maximizes the anomaly scores for the generated boundary samples. This ensures that the anomaly detection model effectively distinguishes between normal and boundary samples by increasing the score difference between the two categories.

Finally, the discriminator D is trained together with the generator G using the original GAN objective from Equation 2, with the corresponding loss $\mathcal{L}_D$ defined as:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim P_{in}(x)} \big[\log D(x) \big] - \mathbb{E}_{z \sim P_{pri}(z)} \big[\log(1 - D(G(z))) \big] \tag{8}$$

This adversarial training framework improves the model's ability to distinguish between normal and anomalous data by iteratively refining both the anomaly detector and the generator. In the next section, we introduce an ensemble-based approach for uncertainty estimation, which serves as a specific realization of this framework.

3.2.3. Ensemble-Based Uncertainty Estimation

Deep ensemble learning has demonstrated strong performance in uncertainty estimation tasks. To implement the framework described in Section

3.2.2, we employ an ensemble-based approach to evaluate uncertainty and guide the generation of effective boundary samples. This section details the construction of the ensemble model, the computation of anomaly scores and uncertainty, and the overall training procedure.

The ensemble anomaly detector is constructed using DeepSAD (Deep Semi-supervised Anomaly Detection) [41]. DeepSAD is selected as the base model due to its ability to incorporate both normal samples and generated boundary samples during training and the loss function of DeepSAD is identical to Equation 7. Other anomaly detection methods capable of utilizing both normal and anomalous samples can also serve as submodels, providing flexibility in the ensemble framework.

DeepSAD maps input data into a representation space using a neural network $\phi_w : X \to Z$, parameterized by w. In this space, the goal is to cluster the representations of normal data points near the center of a hypersphere, while anomalous data points are mapped outside the hypersphere. The anomaly score for a sample x is computed as:

$$s(x) = \|\phi_w(x) - c\|^2 \tag{9}$$

where c is the hypersphere center, determined as the mean of normal data in the representation space. Normal samples are mapped close to c, while anomalous samples are mapped farther away.

DeepSAD assumes that all normal samples cluster around a single center c, which may not hold for complex normal data distributions. However, this issue can also be solved by ensemble learning. The ensemble model in our method is constructed as following: First, the normal dataset is partitioned into K clusters using the K-means algorithm. Then, a separate DeepSAD model is trained for each cluster, resulting in K submodels. Finally, the ensemble anomaly detector combines the outputs of these submodels. The overall anomaly score for a sample x is defined as:

$$A(x) = 1/K \sum_{k=1}^{K} A_k(x) = 1/K \sum_{k=1}^{K} \|\phi_{w_k}(x) - c_k\|^2 \tag{10}$$

This ensemble structure enables the model to effectively evaluate the uncertainty of predictions while handling diverse normal data distributions. When the ensemble model is used to evaluate uncertainty, the generator's losses $\mathcal{L}_{AS}$ (equation 3) and $\mathcal{L}_U$ (equation 4) are redefined as follows:

The ensemble anomaly score loss $\mathcal{L}_{AS}$ is computed as the mean anomaly score across all submodels:

$$\mathcal{L}_{AS} = -\mathbb{E}_{z\sim P_{pri}(z)}\left[1/K\sum_{k=1}^{K} A_k(G(z))\right] \tag{11}$$

The uncertainty loss $\mathcal{L}_U$ measures the standard deviation of the anomaly scores predicted by the submodels:

$$\mathcal{L}_U = \mathbb{E}_{z\sim P_{pri}(z)}\left[\sum_{k=1}^{K}\left|A_k(G(z)) - 1/K\sum_{k=1}^{K} A_k(G(z))\right|\right] \tag{12}$$

Here, A_k denotes the k-th submodel of the ensemble model A. These terms ensure that the generated samples lie in the boundary regions of normal data, have high anomaly scores, and are consistently classified as boundary points by all submodels. This approach prevents overestimation of normality by a single anomaly detector and ensures the quality of generated boundary samples. In addition, the anomaly detector A now comprises an ensemble of submodels, each updated by computing the loss in Equation 7.

By utilizing the ensemble model, we implement the adversarial training framework described in Section 3.2.2. We refer to this method as the Boundary-Aware Adversarial Ensemble Model (BAEM). The training process alternates between optimizing the generator and refining the ensemble anomaly detector. The complete training procedure can be summarized as follows:

(1) Cluster Initialization.
Apply the K-means algorithm to partition the normal training data D_{in} into K clusters, initializing the centers $c_1, \cdots, c_K$ for the submodels.

(2) Submodel Initialization.
Initialize the submodels $A_1^{init}, \cdots, A_K^{init}$ using the clustered data. Each submodel consists of a fixed center c_k and a mapping network ϕ_{w_k}, which is updated during training.

(3) Alternating Training.

- Generator and Discriminator Training: Fix the ensemble detector A, compute $\mathcal{L}_G$ and $\mathcal{L}_D$, then update the generator G and the discriminator D.

- Training of ensemble detector A: Fix the generator G, leverage both normal training samples and generated boundary samples to compute losses $\{\mathcal{L}_A^k\}_{k=1}^K$, then update each submodel A_k.

The full training process ensures the ensemble detector and generator improve iteratively, resulting in a robust framework for detecting anomalies in complex, multivariate time series data. The complete training procedure is outlined in Algorithm 1.

Algorithm 1 Training Algorithm for BAEM

Input: Normal training dataset D_{in}, prior distribution $P_{pri}(z)$, number of clusters K, number of training iterations T_{total}, T_{gan}, and $T_{detector}$.
Output: Trained ensemble detector A and generator G.

/* **Step 1: Cluster and Submodel Initialization** */

Use K-means clustering algorithm to partition D_{in} into K clusters: $\{D_1, D_2, \cdots, D_K\}$.

Initialize each DeepSAD submodel $\{A_k\}_{k=1}^K$ with the corresponding cluster D_k.

repeat

/* **Step 2: Training Generator** G */

repeat

Sample M normal data points $\{x_1, x_2, \cdots, x_M\}$ from D_{in}.

Sample M latent vectors $\{z_1, z_2, \cdots, z_M\}$ from $P_{pri}(z)$.

Generate M synthetic samples $\{G(z_1), G(z_2), \cdots, G(z_M)\}$.

Compute anomaly scores $\{A_k(G(z_j))\}_{k=1}^K$ for each generated sample $G(z_j)$ using all submodels in the ensemble.

Update the generator G and discriminator D using the generator loss $\mathcal{L}_G$ from Equation (6).

Update the discriminator D using $\mathcal{L}_D$ from Equation (8).

until Reach the target number of generator iterations T_{gan}.

/* **Step 3: Training Ensemble Detector** A */

repeat

Sample M normal samples from each cluster D_i, to get K in-distribution sample sets $\{U_1^+, U_2^+, \cdots, U_K^+\}$, where:

$$U_i^+ = \{(x_j, 0)\}_{j=1}^M, \text{ and } (x_j, 0) \sim D_i.$$

Sample M latent vectors $\{z_1, z_2, \cdots, z_M\}$ from $P_{pri}(z)$.

Generate M synthetic anomaly samples $U^- = \{(G(z_j), 1)\}_{j=1}^M$.

Merge normal and anomaly samples to form K new training sets for the submodels:

$$U_i^{new} = U_i^+ \cup U^-, \text{ for } i = 1, 2, \cdots, K.$$

Update the ensemble submodels $\{A_k\}_{k=1}^K$ by corresponding training sets $\{U_i^{new}\}_{i=1}^K$ and the loss $\mathcal{L}_A^k$ from Equation (7).

until Reach the target number of detector iterations $T_{detector}$.

until Reach the total number of iterations T_{total}.

Return: Trained generator G and ensemble detector A.

4. Evaluation

4.1. Evaluation Datasets

To validate the effectiveness of the proposed anomaly detection approach, we conducted experiments across diverse scenarios, including safety-critical systems and common time-series anomaly detection tasks. Below, we provide a detailed description of the datasets used in our evaluation:

- **TLM Dataset**[42] is a UAV (Unmanned Aerial Vehicle) anomaly detection dataset developed in a software-in-the-loop simulation environment, containing logs of UAV anomalies. Four common UAV faults—GPS fault, accelerometer fault, engine fault, and Remote-Control System fault were simulated in the environment. For our experiments, we used the RATE data from this dataset for evaluation.

- **SWaT Dataset**[43] contains data from a scaled-down version of a real-world industrial water treatment plant. It includes 11 days of data, representing both normal and attack states. For the first seven days, the system operated normally, but in the remaining days, it experienced cyber and physical attacks. The dataset captures physical attributes of the plant, water treatment processes, and network traffic logs from the test platform, making it well-suited for evaluating anomaly detection in industrial systems.

- **GHL Collection**[44] is a large collection of simulated multivariate time series datasets from a realistically modeled diesel plant. These datasets were generated with network attacks to simulate plant malfunctions. The collection includes a training dataset with 1.5 million anomaly-free time steps and 48 test datasets, each averaging about

200K time steps. The test sets feature two types of network attacks: unauthorized changes to the maximum tank level and heating tank temperature. Each time step consists of 19 variables, including sensor data and control signals. Our experiments evaluated all 48 test datasets to assess anomaly detection performance comprehensively.

- **SMD Dataset**[11] is a 5-week dataset collected from a large internet company. It contains 39-dimensional trajectory data from 28 machines, and has separate training and testing data for each machine. For this study, we selected four complex sequences from the dataset to evaluate our method's performance.

- **Metro Dataset**[45] contains traffic volume data over 48,204 time steps, spanning from 2012 to 2018. We applied preprocessing methods from previous studies [46] to generate anomaly labels for each time step. After preprocessing, each time step includes five variables: temperature, holiday status, hourly rainfall and snowfall, and cloud cover percentage.

We applied preprocessing methods from the existing literature [46] to the datasets mentioned above except TLM dataset. Then time-domain features were extracted as described in Section 3.2.1. To ensure reproducibility and minimize randomness, all experiments were conducted using three different random seeds. For the GHL and SMD datasets, we directly used their predefined training and testing splits. For the TLM, SWaT, and Metro datasets, we used 80% of the normal data for training and the remaining 20% of normal and anomaly data for evaluation.

4.2. Experimental Settings

4.2.1. Evaluation Metrics

In anomaly detection tasks, commonly used threshold-independent evaluation metrics include the Receiver Operating Characteristic (ROC) curve [47] and the Precision-Recall (PR) curve [48]. Since these curves do not provide a single numeric performance measure, the areas under the curves—AUROC (Area Under the ROC Curve) and AUPR (Area Under the PR Curve)—are often computed to quantitatively assess model performance. In our experiments, we employed these two metrics, where higher values indicate better anomaly detection performance. Both metrics are bounded within the range ($[0, 1]$ as their axes represent ratios.

For evaluation, we chose not to use the Point Adjustment strategy employed in prior studies [49, 50, 11]. This strategy considers an entire anomalous segment as detected if any point within the real anomalous segment is labeled as anomalous. Additionally, it adjusts the anomaly score of all points in a detected segment to the highest score within that segment. While this approach simplifies evaluation, we believe it is overly lenient and can artificially inflate performance metrics by leveraging ground truth information that the model should not have access to. To maintain a fair and realistic evaluation, we directly used the raw anomaly scores predicted by the model and compared them to the ground truth labels without any post-processing. This stricter evaluation approach avoids overestimating model performance but may result in slightly lower scores compared to some previously reported results in the literature.

4.2.2. Competing Methods and Parameter Settings

We compared the proposed BAEM method against a diverse set of mainstream time-series anomaly detection approaches, covering five major categories: (i) One-class classification-based methods: OCSVM [51], DeepSVDD [26], and COUTA [52], (ii) Probability-based methods: Omni [11], ECOD [23], TAnoGAN [53], OCAN [22], and FenceGAN (FGAN) [31], (iii) Reconstruction-based methods: AE [54], Telemanom [55], and Anomaly Transformer (AnoTran) [56], (iv) Density-based methods: LOF [57], (v) Ensemble detectors: Deep Isolation Forest (DIF) [58]. Among these methods, OCAN and FenceGAN are the most similar to our approach, as both use adversarial training to generate boundary samples from distributions of normal data. Additionally, OCSVM, ECOD, DIF, and LOF were implemented using the PyOD library [59]. The implementations of AE, Omni, TAnoGAN, and Telemanom were sourced from an evaluation study [60], while the remaining methods were based on the original code provided by their respective authors. We made every effort to fine-tune the parameters of all methods to achieve their optimal performance.

In all experiments, our ensemble model consisted of 7 submodels, corresponding to the number of clusters in the training data. Each submodel was constructed using an MLP network without bias terms and we employed unbounded activation functions to prevent the hypersphere collapse phenomenon [41]. For the Metro dataset, we modified the original GAN framework to train the anomaly generator. For more complex datasets, we adopted WGAN-GP as the foundation for constructing the anomaly gener-

ator, ensuring improved generative performance. In all cases, the generators were also implemented using MLP networks. During the alternating training process of the base anomaly detectors and the anomaly sample generator, we set both T_{gan} and $T_{detector}$ in Algorithm 1 to 1. This configuration ensured that the detectors and the generator were updated at every iteration.

4.3. Experimental Results

We evaluated the proposed anomaly detection model, BAEM, against 13 baseline methods across five diverse datasets. The results are summarized in Table 1 and Table 2.

Across all datasets listed in Table 1 and Table 2, our method outperforms the baselines in terms of AUROC and AUPR, with reasonable standard deviations, which demonstrates significant performance advantages. All experimental results are reported as percentages, with the best values highlighted in bold. Statistical results for each method were computed using three random seeds. The AUROC and AUPR comparisons between the proposed BAEM and baselines across various datasets are presented as mean ± standard deviation.

Table 1: The comparison of AUROC metrics (mean ± standard deviation) between the proposed method BAEM and 13 methods across five categories on 5 datasets.

Methods	TLM	SWaT	GHL	SMD	Metro
OCSVM	68.39±0.00	66.21±0.00	85.63±0.00	77.18±0.00	60.24±0.00
DeepSVDD	73.15±0.84	76.13±6.50	74.59±7.49	66.27±2.96	52.37±7.56
COUTA	73.80±1.32	85.09±0.24	70.72±7.15	62.27±3.15	76.45±4.49
Omni	64.57±4.24	82.82±3.45	60.28±12.00	63.86±2.10	65.51±4.23
ECOD	69.08±0.00	86.12±0.00	81.66±0.00	76.12±0.00	53.85±0.00
TAnoGAN	62.20±1.47	50.02±0.01	68.29±7.26	58.22±2.76	56.14±4.02
OCAN	64.79±1.36	60.07±14.24	78.82±20.94	65.55±3.34	74.42±9.83
Fence GAN	68.31±2.20	81.28±0.09	74.39±4.37	69.10±0.29	73.22±2.17
AE	70.70±0.62	74.22±0.13	59.27±18.09	73.46±1.01	63.15±10.25
Telemanom	75.77±1.78	78.47±0.01	68.88±2.17	73.75±2.16	N/A
AnomTran	67.48±0.62	85.69±0.01	79.13±0.11	**82.04±0.76**	53.34±2.61
LOF	73.72±0.00	83.86±0.00	87.18±0.00	58.67±0.00	55.60±0.00
DIF	69.04±0.43	**87.05±0.25**	66.80±1.28	73.75±2.16	56.75±3.32
BAEM	**76.10±0.89**	86.70±0.69	**93.52±1.94**	81.54±2.41	**81.09±0.06**

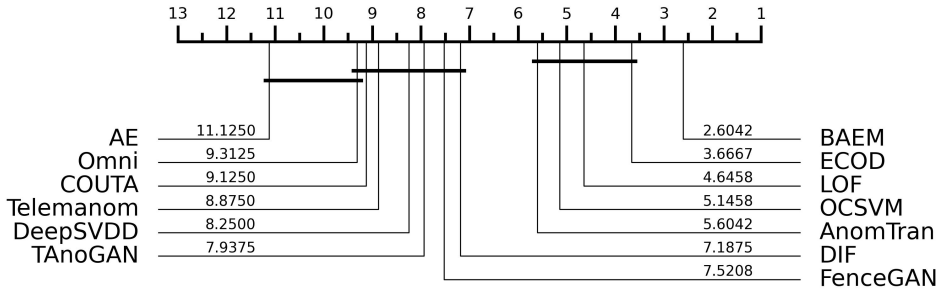
Table 2: The comparison of AUPR metrics (mean ± standard deviation) between the proposed method BAEM and 13 methods across five categories on 5 datasets.

Methods	TLM	SWaT	GHL	SMD	Metro
OCSVM	92.62±0.00	17.47±0.00	8.87±0.00	46.79±0.00	0.79±0.00
DeepSVDD	93.85±0.14	38.68±15.17	5.77±2.95	33.32±4.93	0.68±0.12
COUTA	93.85±0.26	69.97±0.95	1.38±0.73	35.41±4.06	1.98±0.35
Omni	92.48±0.93	38.41±9.79	5.01±4.32	23.12±4.64	1.07±0.37
ECOD	92.19±0.00	75.00±0.00	3.60±0.00	42.17±0.00	0.64±0.00
TAnoGAN	92.20±0.20	12.28±0.01	1.35±0.45	10.05±0.66	0.84±0.02
OCAN	91.28±0.44	27.67±21.77	3.86±4.61	17.47±3.38	**2.52±1.62**
Fence GAN	77.06±1.31	9.45±0.06	0.52±0.11	19.51±0.06	0.40±0.01
AE	93.11±0.12	23.02±0.04	0.77±0.27	52.07±1.37	1.00±0.32
Telemanom	**94.58±0.40**	65.03±1.11	1.28±0.08	38.44±7.14	N/A
AnomTran	92.42±0.12	75.47±0.01	1.26±0.01	52.16±1.10	0.65±0.04
LOF	92.98±0.00	64.59±0.00	7.07±0.00	8.62±0.00	0.80±0.00
DIF	92.66±0.19	75.73±0.41	1.25±0.27	38.44±7.14	0.95±0.05
BAEM	94.26±0.31	**77.28±0.98**	**12.46±5.06**	**57.95±1.87**	1.62±0.01

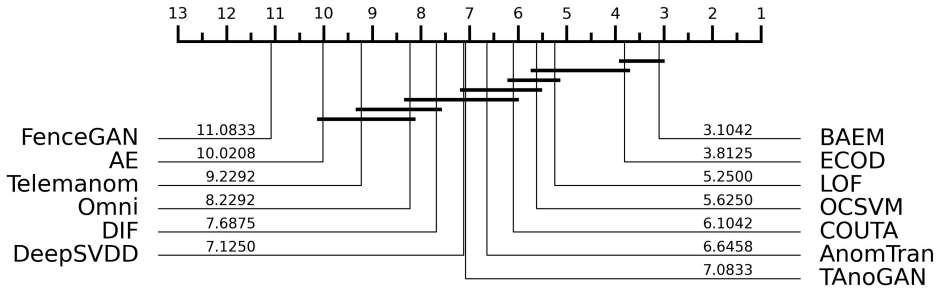
Statistical Analysis. To further validate the performance advantages of BAEM, we conducted statistical tests. The Friedman test was used to examine whether there were significant differences in the performance rankings of all methods across multiple datasets. The null hypothesis of the Friedman test assumes no significant differences between methods, which means the average rankings of the methods across datasets should be similar. Given the extensive training data and 48 test datasets in the GHL Collection, we used the average rankings of the methods on this dataset as input for the Friedman test. The results yielded a p-value of $p = 0.00$, which is less than the significance level of 0.05, and leads to the rejection of the null hypothesis. This indicates that there are statistically significant differences between the methods.

To further investigate pairwise differences between methods, we utilized the Wilcoxon signed-rank test. This non-parametric test evaluates whether the performance differences between two methods are statistically significant. Using the aeon toolkit [61], we generated critical difference diagrams for both AUROC and AUPR, as shown in Figure 4. From Figure 4, it can be observed at the 95% confidence level, BAEM significantly outperforms all other methods except ECOD.

Based on the results presented above, we confirm that the performance



(a) AUROC as the evaluation metric.



(b) AUPR as the evaluation metric.

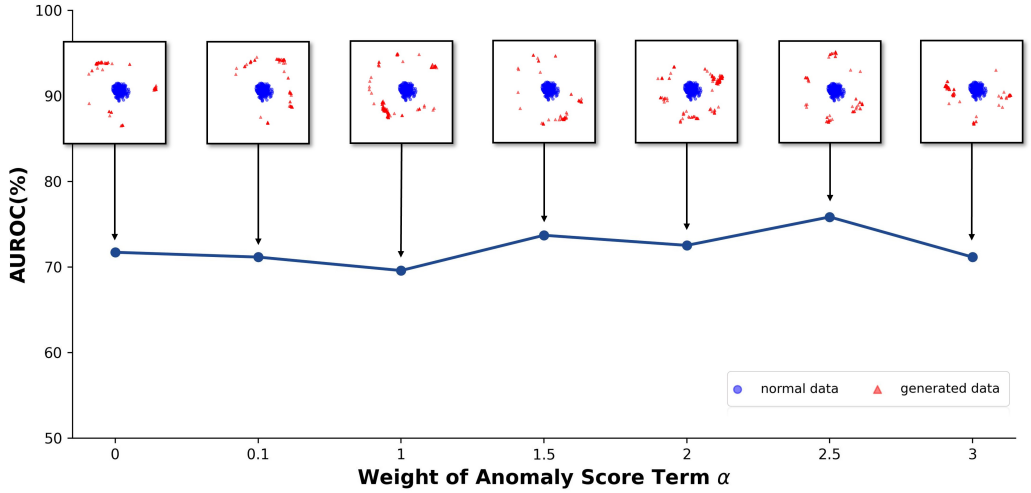
Figure 4: Critical difference diagram for the Wilcoxon signed-rank test. The numbers represent the average rankings of each method across the 48 datasets in the GHL Collection. Lower average rankings indicate better methods. The horizontal lines in the diagram highlights methods that do not show significant differences in the metric.

improvements achieved by BAEM are statistically significant. These findings demonstrate the strength of our method in addressing diverse anomaly detection tasks.

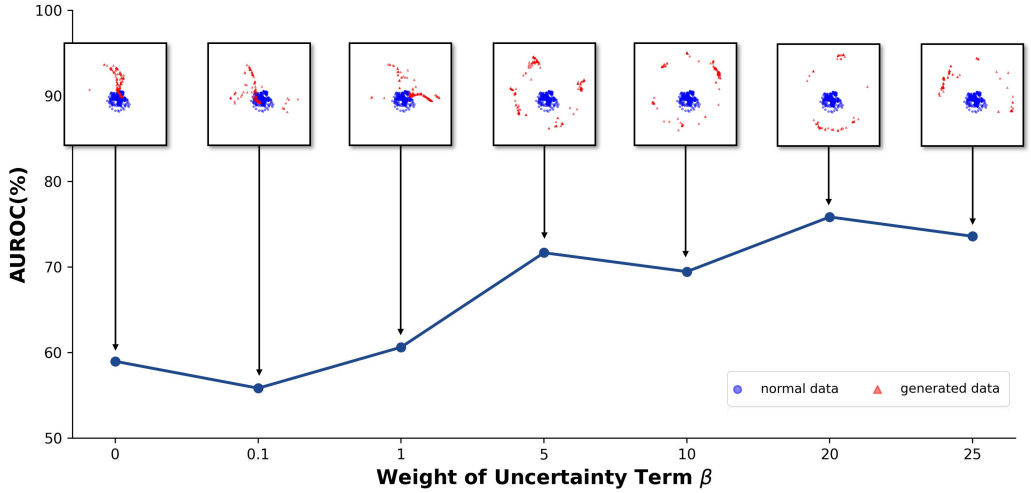
4.4. Experiments on Parameter Setting

To evaluate the sensitivity of BAEM to various parameters, we conducted tests with different configurations. First, we focused on two key parameters that control the generation of boundary samples:

- α: The weight of the anomaly score loss $\mathcal{L}_{AS}$ in the generator loss $\mathcal{L}_{GAN}$.
- β: The weight of the uncertainty loss $\mathcal{L}_{U}$ in the generator loss $\mathcal{L}_{GAN}$.



(a) Experiments with different α values.



(b) Experiments with different β values.

Figure 5: Results of the AUROC performance and visualization of generated boundary samples under different α and β settings. Blue circles represent normal samples, while red triangles represent generated boundary samples. The results highlight that β plays a more critical role in controlling the generation of boundary samples.

Our experiments indicate the weight of the uncertainty term β, plays a more critical role in controlling boundary sample generation compared to the anomaly score term α, which has been commonly used in prior work.

We performed experiments on the TLM dataset to analyze the impact of

these two parameters. To isolate the effects of each parameter, we adjusted only one parameter at a time while keeping the other fixed: For α, we fixed $\beta = 20$. And for β, we fixed $\alpha = 2.5$. Additionally, we set the total number of iterations $T_{total} = 70$ and used Multidimensional Scaling (MDS) to project the generated boundary samples into a 2D space for visualization. The results are shown in Figure 5.

According to Figure 5(a), when β is fixed at a reasonable value, variations in α have minimal impact on the location of boundary samples, which results in minor changes to anomaly detection performance. In contrast, Figure 5(b) shows that the weight of the uncertainty term β significantly affects the generation of boundary samples. When β is set to a low value, the inaccurate predictions of the anomaly detector seriously influence the generator and results in the generated samples similar to normal samples. This finally disrupts the detector's training process and impacts performance. As β increases, the generated samples become more distinguishable from normal samples, which improves the detector's performance. These findings validate the motivation for incorporating the uncertainty loss into the generation of boundary samples.

In addition, we also analyzed the sensitivity of BAEM to other key training parameters:

- η: The initial learning rate of the Adam optimizer used during generator training.
- H: The dimensionality of the generator's latent space.
- K: The number of submodels in the ensemble detector.

These parameters were tested across a wide range of values, and the results are shown in Figure 6.

The results indicate that The latent space H dimensionality significantly affects the model's ability to generate high-quality boundary samples. A properly tuned H ensures that the generator captures sufficient complexity while avoiding overfitting. The number of submodels K in the ensemble detector greatly influences performance. Notably, the performance does not improve significantly when the number of submodels is high. Our experiments suggest that approximately seven submodels are sufficient to achieve robust performance.

In addition, on the TLM dataset, the model's performance metrics are more sensitive to changes in the number of submodels K. This is likely due

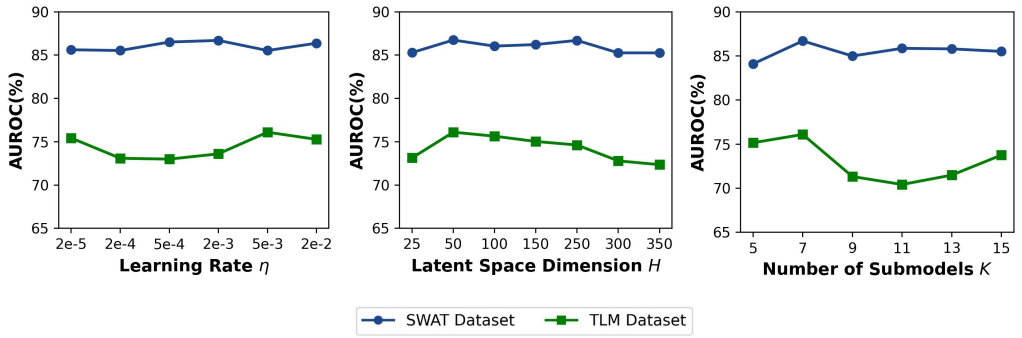


Figure 6: The sensitivity test results of η, H and K. The results show AUROC performance across different settings for the SWaT and TLM datasets. Different colors correspond to different datasets.

to the smaller size of the TLM dataset, which may limit the effectiveness of training some submodels. To mitigate this issue, data augmentation techniques could be employed to expand the dataset and enhance the accuracy of the model.

4.5. Ablation Study

In this subsection, we analyze the contributions of key components in BAEM by comparing it with its four variants. A detailed ablation study was conducted to evaluate the effectiveness of different loss function components in BAEM. Specifically, we employed various loss function configurations to generate boundary data, which were then used to train anomaly detectors. The final performance of these detectors was compared to assess the impact of each component.

The following configurations were evaluated:

- **DeepSAD**: The original DeepSAD model without any ensemble or generated boundary data.
- **Ensemble DeepSAD**: DeepSAD enhanced with an ensemble approach but without using generated boundary data.
- **BAEM (w/o Anomaly Score Loss)**: BAEM without the anomaly score loss component ($\mathcal{L}_{AS}$).
- **BAEM (w/o Uncertainty Loss)**: BAEM without the uncertainty loss component ($\mathcal{L}_{U}$).

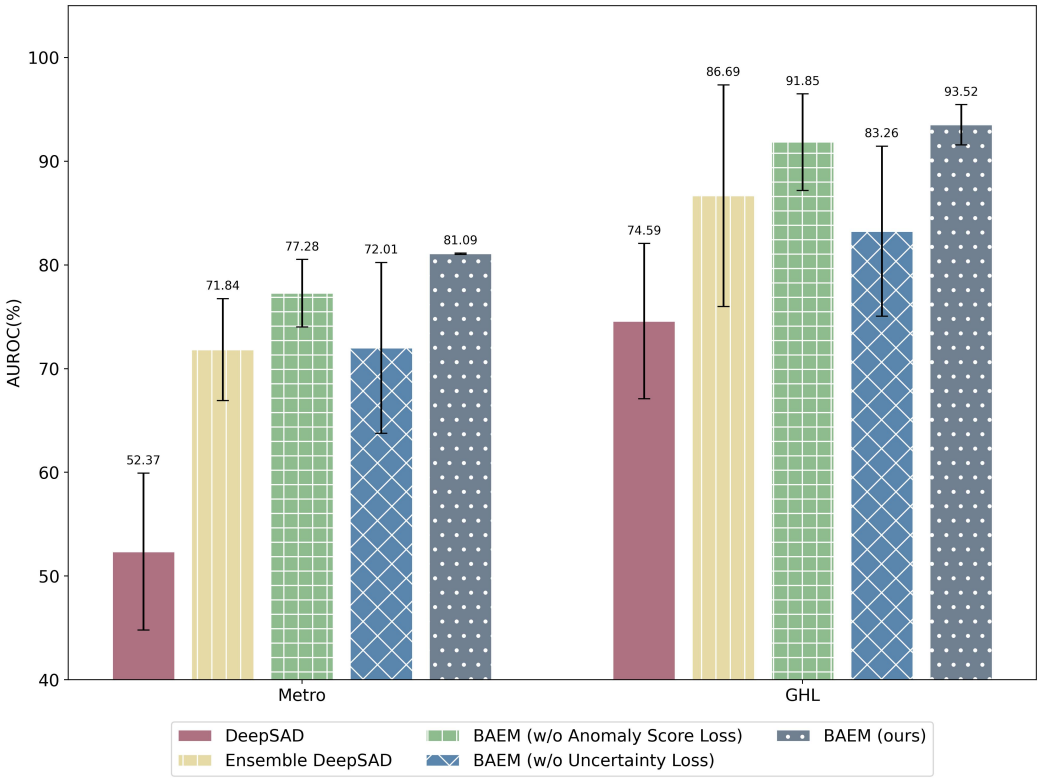


Figure 7: Comparison of performance for BAEM and its 4 variants on Metro and GHL datasets. The bar chart uses distinct colors and patterns to represent the different configurations. The x-axis corresponds to the evaluation metric (AUROC), while the y-axis represents the performance score (ranging from 0% to 100%). Error bars indicate the standard deviation of the performance, and numeric labels on top of each bar show the mean values.

- **BAEM (ours)**: The complete BAEM model with all loss function components.

Figure 7 presents the comparison of AUROC performance for the above configurations on the Metro and GHL datasets.

On both datasets, the complete BAEM model achieves the highest performance, highlighting the importance of incorporating both the anomaly score loss ($\mathcal{L}_{AS}$) and uncertainty loss ($\mathcal{L}_U$) into the framework. Besides, the Ensemble DeepSAD configuration significantly outperforms the original DeepSAD, which validates the ensemble approach as an effective enhancement for time-series anomaly detection tasks.

Furthermore, the comparison between BAEM (w/o Uncertainty Loss) with BAEM (w/o Anomaly Score Loss) reveals that removing the uncertainty loss ($\mathcal{L}_U$) causes a more significant drop in performance. This underscores the vital role of the uncertainty loss in improving the model's generalization ability. The results align with the findings in Section 4.4, which further validate the uncertainty loss' critical role in ensuring high-quality boundary sample generation.

5. Conclusion

This paper proposes the Boundary-Aware Adversarial Ensemble Model (BAEM) to address the challenges of multivariate time series anomaly detection in safety-critical systems. BAEM effectively tackles the scarcity of anomalous data and the complexity of anomaly patterns by generating boundary samples based on prior knowledge and refining the detection model through adversarial training. An ensemble approach ensures that generated boundary samples exhibit low uncertainty, improving model reliability and robustness.

Extensive experiments on real-world and benchmark datasets demonstrate that BAEM outperforms 13 mainstream methods, achieving significant improvements in AUROC and AUPR metrics. Notably, we identify uncertainty as a more critical factor than anomaly score for boundary sample generation, providing reliable supervisory signals that enhance detection performance. This insight offers a new direction for improving anomaly detection methods.

Future work will focus on refining the boundary sample generation process and exploring BAEM's application in dynamic environments, such as domain adaptation scenarios. Additionally, leveraging uncertainty estimation in real-world applications can improve model trustworthiness, balancing safety and operational efficiency. These efforts aim to further establish BAEM as a reliable solution for anomaly detection in safety-critical systems.

Acknowledgements

This work is partially supported by National Natural Science Foundation Integration Project (No.U22B6001) and National Natural Science Foundation of China (6247072715).

References

[1] Y.-J. Park, S.-K. S. Fan, C.-Y. Hsu, A review on fault detection and process diagnostics in industrial processes, Processes 8 (9) (2020). doi:10.3390/pr8091123.

[2] D. Bogdoll, M. Nitsche, J. M. Zöllner, Anomaly detection in autonomous driving: A survey, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 4487–4498. doi:10.1109/CVPRW56347.2022.00495.

[3] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, Q. Zhang, Time-series anomaly detection service at microsoft, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 3009–3017. doi:10.1145/3292500.3330680.

[4] K. Shaukat, T. M. Alam, S. Luo, S. Shabbir, I. A. Hameed, J. Li, S. K. Abbas, U. Javed, A review of time-series anomaly detection techniques: A step to future perspectives, in: K. Arai (Ed.), Advances in Information and Communication, Springer International Publishing, Cham, 2021, pp. 865–877.

[5] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A. K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mechanical Systems and Signal Processing 138 (2020) 106587. doi:https://doi.org/10.1016/j.ymssp.2019.106587.
URL https://www.sciencedirect.com/science/article/pii/S0888327019308088

[6] Z. Gao, C. Cecati, S. X. Ding, A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches, IEEE Transactions on Industrial Electronics 62 (6) (2015) 3768–3774. doi:10.1109/TIE.2015.2419013.

[7] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE 109 (5) (2021) 756–795. doi:10.1109/JPROC.2021.3052449.

[8] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 427–436. doi:10.1109/CVPR.2015.7298640.

[9] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. Van Den Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1705–1714. doi:10.1109/ICCV.2019.00179.

[10] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.
URL https://openreview.net/forum?id=BJJLHbb0-

[11] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, 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, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2828–2837. doi:10.1145/3292500.3330672.

[12] Y. Liu, X. Li, X. Zhang, L. Fan, X. Chen, B. Gong, Imbalanced deep transfer network for fault diagnosis of high-speed train traction motor bearings, Knowledge-Based Systems 293 (2024) 111682. doi:https://doi.org/10.1016/j.knosys.2024.111682.

[13] Z. Liang, H. Wang, X. Ding, T. Mu, Industrial time series determinative anomaly detection based on constraint hypergraph, Knowledge-Based Systems 233 (2021) 107548. doi:https://doi.org/10.1016/j.knosys.2021.107548.

[14] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, C. Fookes, Deep learning for medical anomaly detection – a survey, ACM Comput. Surv. 54 (7) (jul 2021). doi:10.1145/3464423.

[15] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, G. Troster, Wearable assistant for parkinson's disease patients

with the freezing of gait symptom, IEEE Transactions on Information Technology in Biomedicine 14 (2) (2010) 436–446. doi:10.1109/TITB.2009.2036165.

[16] S. Latif, Z. Idrees, Z. Zou, J. Ahmad, Drann: A deep random neural network model for intrusion detection in industrial iot, in: 2020 International Conference on UK-China Emerging Technologies (UCET), 2020, pp. 1–4. doi:10.1109/UCET51115.2020.9205361.

[17] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: an ensemble of autoencoders for online network intrusion detection, arXiv preprint arXiv:1802.09089 (2018).

[18] V. Q. Nguyen, L. T. Ngo, L. M. Nguyen, V. H. Nguyen, N. Shone, Deep clustering hierarchical latent representation for anomaly-based cyber-attack detection, Knowledge-Based Systems 301 (2024) 112366. doi:https://doi.org/10.1016/j.knosys.2024.112366.

[19] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis, 2014, pp. 4–11.

[20] O. I. Provotar, Y. M. Linder, M. M. Veres, Unsupervised anomaly detection in time series using lstm-based autoencoders, in: 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), 2019, pp. 513–517. doi:10.1109/ATIT49449.2019.9030505.

[21] T. Kieu, B. Yang, C. S. Jensen, Outlier detection for multidimensional time series using deep neural networks, in: 2018 19th IEEE international conference on mobile data management (MDM), IEEE, 2018, pp. 125–134.

[22] P. Zheng, S. Yuan, X. Wu, J. Li, A. Lu, One-class adversarial nets for fraud detection, Proceedings of the AAAI Conference on Artificial Intelligence 33 (01) (2019) 1286–1293. doi:10.1609/aaai.v33i01.33011286. URL https://ojs.aaai.org/index.php/AAAI/article/view/3924

[23] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, G. H. Chen, Ecod: Unsupervised outlier detection using empirical cumulative distribution functions, IEEE Transactions on Knowledge and Data Engineering 35 (12) (2023) 12181–12193. doi:10.1109/TKDE.2022.3159580.

[24] Y. Shi, B. Wang, Y. Yu, X. Tang, C. Huang, J. Dong, Robust anomaly detection for multivariate time series through temporal gcns and attention-based vae, Knowledge-Based Systems 275 (2023) 110725. doi:https://doi.org/10.1016/j.knosys.2023.110725.

[25] Y. Yao, J. Ma, Y. Ye, Kfreqgan: Unsupervised detection of sequence anomaly with adversarial learning and frequency domain information, Knowledge-Based Systems 236 (2022) 107757. doi:https://doi.org/10.1016/j.knosys.2021.107757.

[26] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 4393–4402.

[27] S. M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning, Pattern Recognition 58 (2016) 121–134. doi:https://doi.org/10.1016/j.patcog.2016.03.028.

[28] P. Perera, V. M. Patel, Learning deep features for one-class classification, IEEE Transactions on Image Processing 28 (11) (2019) 5450–5463. doi:10.1109/TIP.2019.2917862.

[29] D. Hendrycks, M. Mazeika, T. Dietterich, Deep anomaly detection with outlier exposure, in: International Conference on Learning Representations, 2019.
URL https://openreview.net/forum?id=HyxCxhRcY7

[30] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, J. Wang, Outlier exposure with confidence control for out-of-distribution detection, Neurocomputing 441 (2021) 138–150. doi:https://doi.org/10.1016/j.neucom.2021.02.007.

[31] P. C. Ngo, A. A. Winarto, C. K. L. Kou, S. Park, F. Akram, H. K. Lee, Fence gan: Towards better anomaly detection, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 141–148. doi:10.1109/ICTAI.2019.00028.

[32] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, X. He, Generative adversarial active learning for unsupervised outlier detection, IEEE Transactions on Knowledge and Data Engineering 32 (8) (2020) 1517–1528. doi:10.1109/TKDE.2019.2905606.

[33] P. Schlachter, Y. Liao, B. Yang, Deep one-class classification using intra-class splitting, in: 2019 IEEE Data Science Workshop (DSW), 2019, pp. 100–104. doi:10.1109/DSW.2019.8755576.

[34] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley interdisciplinary reviews: data mining and knowledge discovery 8 (4) (2018) e1249.

[35] L. Breiman, Bagging predictors, Machine learning 24 (1996) 123–140.

[36] T. Hastie, S. Rosset, J. Zhu, H. Zou, Multi-class adaboost, Statistics and its Interface 2 (3) (2009) 349–360.

[37] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 785–794.
URL https://doi.org/10.1145/2939672.2939785

[38] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, S. Nowozin, J. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, Advances in neural information processing systems 32 (2019).

[39] P. Boonyakitanont, A. Lek-uthai, K. Chomtho, J. Songsiri, A review of feature extraction and performance evaluation in epileptic seizure detection using eeg, Biomedical Signal Processing and Control 57 (2020) 101702. doi:https://doi.org/10.1016/j.bspc.2019.101702.
URL https://www.sciencedirect.com/science/article/pii/S1746809419302836

[40] A. O. Ige, M. H. Mohd Noor, A survey on unsupervised learning for wearable sensor-based activity recognition, Applied Soft Computing 127 (2022) 109363. doi:https://doi.org/10.1016/j.asoc.2022.109363.
URL https://www.sciencedirect.com/science/article/pii/S1568494622005191

[41] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft, Deep semi-supervised anomaly detection, in: International Conference on Learning Representations, 2020.
URL https://openreview.net/forum?id=HkgH0TEYwH

[42] T. Yang, Y. Lu, H. Deng, J. Chen, X. Tang, Acquisition and processing of uav fault data based on time line modeling method, Applied Sciences 13 (7) (2023). doi:10.3390/app13074301.
URL https://www.mdpi.com/2076-3417/13/7/4301

[43] J. Goh, S. Adepu, K. N. Junejo, A. Mathur, A dataset to support research in the design of secure water treatment systems, in: G. Havarneanu, R. Setola, H. Nassopoulos, S. Wolthusen (Eds.), Critical Information Infrastructures Security, Springer International Publishing, Cham, 2017, pp. 88–99.

[44] P. Filonov, A. Lavrentyev, A. Vorontsov, Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model, arXiv preprint arXiv:1612.06676 (2016).

[45] J. Hogue, Metro Interstate Traffic Volume, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5X60B (2019).

[46] S. Schmidl, P. Wenig, T. Papenbrock, Anomaly detection in time series: a comprehensive evaluation, Proc. VLDB Endow. 15 (9) (2022) 1779–1797.
URL https://doi.org/10.14778/3538598.3538602

[47] C. E. Metz, Basic principles of roc analysis, Seminars in Nuclear Medicine 8 (4) (1978) 283–298. doi:https://doi.org/10.1016/S0001-2998(78)80014-2.

[48] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 233–240.
URL https://doi.org/10.1145/1143844.1143874

[49] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, H. Qiao, Unsupervised anomaly

detection via variational auto-encoder for seasonal kpis in web applications, in: Proceedings of the 2018 World Wide Web Conference, WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, p. 187–196. doi:10.1145/3178876.3185996.
URL https://doi.org/10.1145/3178876.3185996

[50] S. Tuli, G. Casale, N. R. Jennings, Tranad: deep transformer networks for anomaly detection in multivariate time series data, Proc. VLDB Endow. 15 (6) (2022) 1201–1214. doi:10.14778/3514061.3514067.
URL https://doi.org/10.14778/3514061.3514067

[51] L. M. Manevitz, M. Yousef, One-class svms for document classification, J. Mach. Learn. Res. 2 (2002) 139–154.

[52] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, G. Pang, Calibrated one-class classification for unsupervised time series anomaly detection, IEEE Transactions on Knowledge and Data Engineering (2024) 1–14doi:10.1109/TKDE.2024.3393996.

[53] M. A. Bashar, R. Nayak, Tanogan: Time series anomaly detection with generative adversarial networks, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, pp. 1778–1785. doi:10.1109/SSCI47803.2020.9308512.

[54] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, MLSDA'14, Association for Computing Machinery, New York, NY, USA, 2014, p. 4–11. doi:10.1145/2689746.2689747.
URL https://doi.org/10.1145/2689746.2689747

[55] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 387–395. doi:10.1145/3219819.3219845.

[56] J. Xu, H. Wu, J. Wang, M. Long, Anomaly transformer: Time series anomaly detection with association discrepancy, in: International Conference on Learning Representations, 2022.
URL https://openreview.net/forum?id=LzQQ89U1qm_

[57] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, SIGMOD Rec. 29 (2) (2000) 93–104. doi:10.1145/335191.335388.
URL https://doi.org/10.1145/335191.335388

[58] H. Xu, G. Pang, Y. Wang, Y. Wang, Deep isolation forest for anomaly detection, IEEE Transactions on Knowledge and Data Engineering 35 (12) (2023) 12591–12604. doi:10.1109/TKDE.2023.3270293.

[59] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, Journal of Machine Learning Research 20 (96) (2019) 1–7.
URL http://jmlr.org/papers/v20/19-011.html

[60] P. Wenig, S. Schmidl, T. Papenbrock, Timeeval: A benchmarking toolkit for time series anomaly detection algorithms 15 (12) 3678–3681. doi:10.14778/3554821.3554873.

[61] M. Middlehurst, A. Ismail-Fawaz, A. Guillaume, C. Holder, D. Guijo-Rubio, G. Bulatova, L. Tsaprounis, L. Mentel, M. Walter, P. Schäfer, A. Bagnall, aeon: a python toolkit for learning from time series, Journal of Machine Learning Research 25 (289) (2024) 1–10.
URL http://jmlr.org/papers/v25/23-1444.html

Dear Editors:

We the undersigned declare that this manuscript entitled "**Boundary-Aware Adversarial Ensemble Learning for Multivariate Time Series Anomaly Detection**" is original, has not been published before and is not currently being considered for publication elsewhere.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We understand that the Corresponding Author is the sole contact for the Editorial process. He is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs.

Signed by all authors as follows: Pengcheng He, Xiaoyang Tan, Yuehua Cheng

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: