

Article

Enhancing Anomaly Detection in Maritime Operational IoT Time Series Data with Synthetic Outliers

Hyunjoo Kim  and Inwhee Joe * 

Department of Computer Science, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea; hjkim666@hanyang.ac.kr

* Correspondence: iwjoe@hanyang.ac.kr

Abstract: Detecting anomalies in engine and machinery data during ship operations is crucial for maintaining the safety and efficiency of the vessel. We conducted experiments using device data from the maritime industry, consisting of time series records from IoT (Internet of Things) datasets such as cylinder and exhaust gas temperatures, coolant temperatures, and cylinder pressures collected from various sensors on the ship's equipment. We propose data enrichment and validation techniques by generating synthetic outliers through data degradation and data augmentation with a Transformer backbone, utilizing the maritime operational data. We extract a portion of the input data and replace it with synthetic outliers. The created anomaly data are then used to train the model via a self-supervised learning approach. Synthetic outliers are generated using methods such as the arithmetic mean, geometric mean, median, local scale, global scale, and magnitude warping. With our methodology, we achieved a 17.23% improvement in F1 performance compared to existing state-of-the-art methods across five publicly available datasets and actual maritime operational data collected from the industry.

Keywords: anomaly detection; time series; synthetic outlier; outlier generation; maritime operational data; IoT anomaly detection



Citation: Kim, H.; Joe, I. Enhancing Anomaly Detection in Maritime Operational IoT Time Series Data with Synthetic Outliers. *Electronics* **2024**, *13*, 3912. <https://doi.org/10.3390/electronics13193912>

Academic Editors: Zijian Li, Zihan Chen, Jiahui Li and Longyu Zhou

Received: 7 August 2024

Revised: 16 September 2024

Accepted: 28 September 2024

Published: 3 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over 80% of the volume of global trade is transported by ships [1]. Analyzing the engine and machinery sensing data from ships is crucial for enhancing operational efficiency, maintaining safety, and detecting potential issues before they become critical.

IoT-based smart shipping plays a crucial role in enhancing vessel performance [2], ensuring safety, enabling timely intervention, reducing downtime, and preemptively detecting unexpected issues (anomaly detection). This involves monitoring various parameters such as fuel consumption, engine performance, temperature, and humidity [3].

An integrated ship control system (ISCS) typically includes the engine control system, power management system, generator and distribution control system, and propulsion system control. This paper utilizes data collected from the generator and distribution system. The generator and distribution control system is responsible for managing the generation and distribution of power on a ship. It efficiently distributes the electricity produced by the generators to various systems on the ship (such as engines, cooling systems, lighting, communication equipment, etc.), ensuring a stable power supply for the vessel. The dataset managed by the ISCS was transmitted via MQTT and stored in InfluxDB, which specializes in collecting, storing, and analyzing time series data, and was used for data analysis in this study.

We used data recorded every minute from October 2022 to August 2023. We named this dataset the Industrial Maritime Operational Dataset (IMOD). Using this dataset, we improved the performance of existing Transformer-based models and developed a new model for anomaly detection. To improve the performance of the Transformer-based model,

we generated synthetic outliers to detect point-wise anomalies such as global anomalies, contextual anomalies, and pattern-wise anomalies, including shapelet, seasonal, and trend anomalies [4].

Our contributions are as follows:

- Through our outlier synthetic framework, we improved the performance of anomaly detection on time series datasets beyond the current state-of-the-art (SOTA) methods.
- Our methodology improved the F1 performance for the Industrial Maritime Operational Dataset (IMOD), which has an anomaly rate of only 0.03.

In Section 2, we introduce studies related to time series anomaly detection and synthetic outliers. In Section 2.2, we explain our framework for generating synthetic outliers. The Section 3 details the datasets we utilized. In Section 3.2, we describe how we trained our model. Section 3.3 evaluates our proposed anomaly detection method. Finally, in Section 5, we summarize the strengths and weaknesses of our research approach and suggest directions for future studies.

2. Materials and Methods

2.1. Related Work

Time series anomaly detection (TAD). The anomalies in the time series come in various forms, as shown in Figure 1, each with distinct characteristics. Global anomalies manifest as spikes in a time series, denoting points with exceptionally high or low values compared to the series' overall pattern. Contextual anomalies involve deviations from neighboring time points within a defined proximity range, relative to a given context. Seasonal anomalies exhibit unusual seasonality within time series, despite similar shapes and trends. Trend anomalies represent events causing a permanent shift in data toward the mean, resulting in a transition in the time series trend. Shapelet anomalies are identified by subsequences with shapelets or cycles differing from the normal components of the sequence. In essence, an anomaly denotes a deviation from the general distribution of data, whether in the form of a single outlier or a series of observations significantly diverging from the norm.

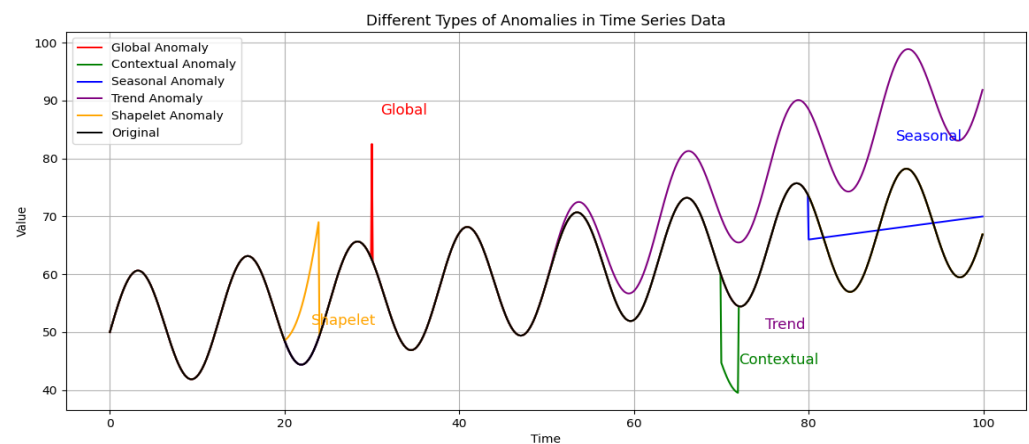


Figure 1. Various types of anomalies in time series data: global anomaly (red), contextual anomaly (green), seasonal anomaly (blue), trend anomaly (purple), shapelet anomaly (orange), and original data (black).

The methodology for anomaly detection in time series can be categorized into traditional machine learning methods, statistical methods, and deep learning methods, among others. Traditional machine learning methods include SVMs, random forests, KNNs, isolation forests, and clustering algorithms, such as DBSCAN and K-means. Deep learning methodologies include LSTM-based (long short-term memory-based) [5–7], CNN-based (convolutional neural network-based) [8–10], and Transformer-based approaches.

Reconstruction-based methodologies include autoencoders [11,12] and VAEs (variational autoencoders) [13,14]. Moreover, there are generative-based anomaly detection methods utilizing the generative capabilities of GANs (generative adversarial networks) [15,16].

Synthetic outlier

Synthetic outliers are artificially generated data points that deviate significantly from the norm. There are studies on various synthetic outlier methods for time series. Basic approaches include window cropping [17,18], window slicing [18], window warping (DTW) [19], flipping [20], and noise injection [8]. Additionally, there are frequency-domain and decomposition methods suggested for augmentation. When focusing on methods that modify the sizes of time series data based on the Y axis, techniques include jittering [17], flipping [20], scaling [17], and magnitude warping [21]. For methods that modify the time axis (X-axis), techniques include permutation [17,19], window slicing [18], time warping [19,22], and window warping [19]. AnomalyBERT [23] uses four types of synthetic outliers—soft replacement, uniform replacement, peak noise, and length adjustment in a Transformer-based model for anomaly detection. Additionally, AnomalyBERT covers five types of anomalies: global point, contextual point, shapelet pattern, seasonal pattern, and trend pattern.

Transformer for TAD

Recently, Transformer-based methodologies have been extensively researched, demonstrating superior performance for TAD. These include Informer [24], which utilizes the ProbSparse self-attention mechanism, AnomalyBERT [23], which employs a data degradation scheme for self-supervised learning, Transformers for multivariate time series anomaly detection [25], which capture both temporal and inter-variable dependencies using an inter-variable attention mechanism, and domain adaptation contrastive learning (DA-CAD) with Transformers [26], which uses contrastive learning, domain adaptation, and anomaly injection.

2.2. Proposed Method

The anomaly detection framework for maritime operational data comprises a synthetic outlier generation module and a Transformer backbone model [23]. As shown in Figure 2, the main module is a synthetic outlier generation module with nine sub-modules and a Transformer backbone. After training, the model is selected with the best F1 score, as seen in Figure 3b. The synthetic outlier generation module consists of the following submodules:

- **Arithmetic mean:** Transforms data points within a specific window by calculating their arithmetic mean. This module generates outliers as shown in Figure 4a.
- **Geometric mean:** Transforms data points within a specific window by calculating their geometric mean. This module generates outliers similar to those shown in Figure 4b.
- **Median:** Transforms data points within a specific window by calculating their median value. This module generates outliers similar to those shown in Figure 5a.
- **Global scaling:** Transforms data points within a specific window by applying global scaling. This module generates outliers similar to those shown in Figure 6a.
- **Local scaling:** Transforms data points within a specific window by applying local scaling. This module generates outliers similar to those shown in Figure 5b.
- **Magnitude warping:** Transforms data points within a specific window by applying magnitude warping. This module generates outliers similar to those shown in Figure 6b.
- **Flip:** Transforms data points within a specific window by applying a horizontal or vertical flip; see [23].
- **Peak nosing:** Transforms data points within a specific window by injecting point-wise random noise; see [23].
- **Random constant value replacing:** Transforms data points within a specific window by choosing random constant values from uniform distribution; see [23].

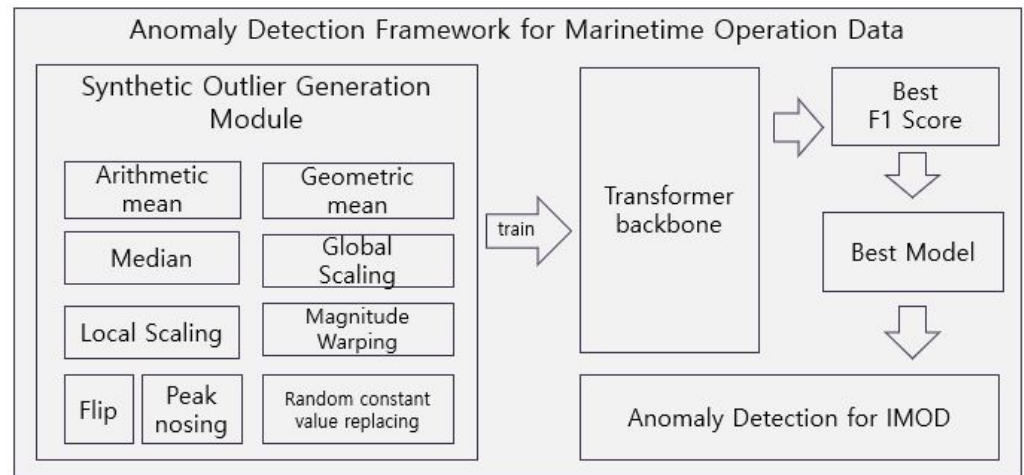
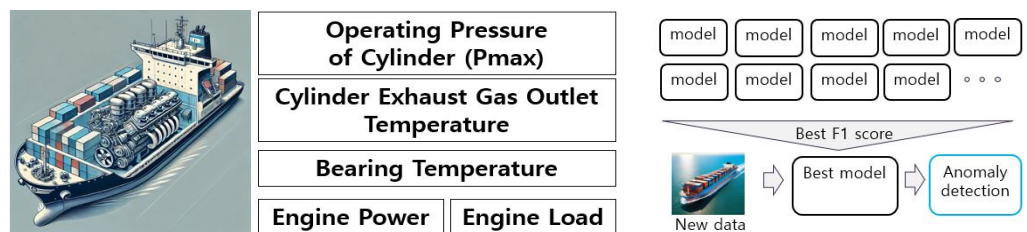


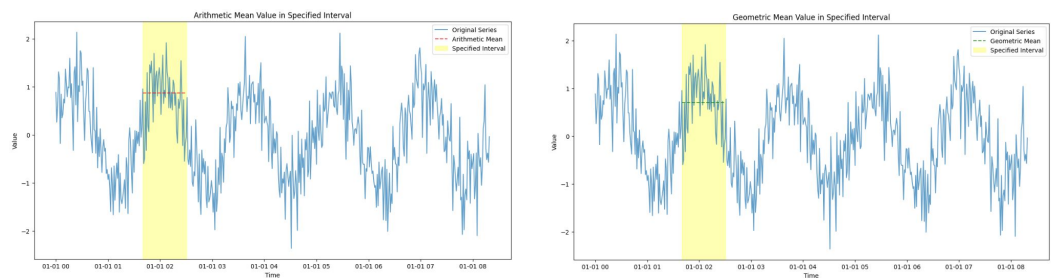
Figure 2. This diagram depicts an anomaly detection framework for maritime operational data. Outliers are generated through the synthetic outlier generation module and used to train the Transformer backbone model, which produces an F1 score. The model with the best score is determined through voting to identify the best model, which is then used for anomaly detection.



(a) Description of key data for ship equipment.

(b) Model selection for AD

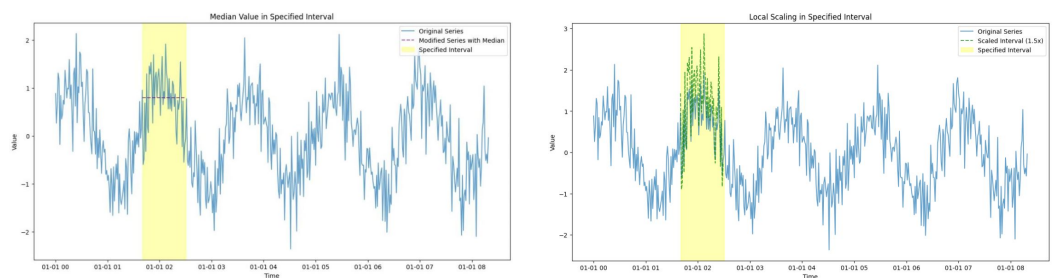
Figure 3. (a) Describes important data from ship equipment related to the engine; (b) details the model selection method used to choose the best model for anomaly detection.



(a) Arithmetic mean

(b) Geometric mean

Figure 4. Synthetic outlier type for anomaly detection using arithmetic mean and geometric mean.



(a) Median

(b) local scaling

Figure 5. Synthetic outlier type for anomaly detection using median and local scaling

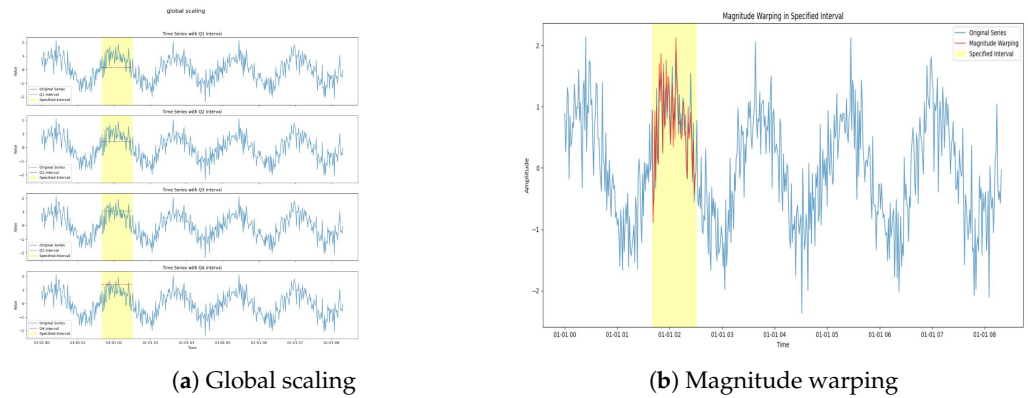


Figure 6. Synthetic outlier type for anomaly detection using global scaling and magnitude warping.

Synthetic outlier generation module.

Let $\mathbf{x} = [x_1, x_2, \dots, x_T]$ be the original time series data. We select an interval $[t_i, t_j]$ where $0 < i < j < T$. The selected sequence $[x_{t_i} : x_{t_j}]$ is transformed into synthetic outliers by the synthetic outlier generation module.

- Arithmetic mean

$$\bar{x} = \frac{1}{j-i+1} \sum_{t=t_i}^{t_j} x_t \tag{1}$$

- Geometric mean

$$\bar{x}_{geo} = \left(\prod_{t=t_i}^{t_j} x_t \right)^{\frac{1}{j-i+1}} \tag{2}$$

- Median

$$\tilde{x} = \begin{cases} x_{(\frac{n+1}{2})} & \text{if } n \text{ is odd, } n = j-i+1 \\ \frac{1}{2} (x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}) & \text{if } n \text{ is even, } n = j-i+1 \end{cases} \tag{3}$$

- Global scaling

$$\tilde{x} = X_k \cdot \text{scaling_factor}, \quad \text{scaling_factor} = \mu_{ij} - \sigma_{ij} \tag{4}$$

- Local scaling

$$\tilde{x} = \begin{cases} 1.5 \cdot X_k & \text{if } Q_1 \leq X_k \leq Q_3 \\ X_k & \text{otherwise} \end{cases} \quad \text{for } k = t_i, t_i + 1, \dots, t_j. \tag{5}$$

- Magnitude warping

$$\tilde{\mathbf{x}} = \mathbf{x} \cdot \mathbf{w}, \quad \mathbf{w} \sim \mathcal{N}(1, \sigma^2) \tag{6}$$

Equation (1) is the arithmetic mean from x_{t_i} to x_{t_j} . This result can be seen in Figure 4a. Equation (2) is the geometric mean from x_{t_i} to x_{t_j} . This result can be seen in Figure 4b. Equation (3) represents the median from t_i to t_j . This result can be seen in Figure 5a. In Equation (4), μ_{ij} denotes the mean and σ_{ij} denotes the standard deviation of the sequence $[X_{t_i} : X_{t_j}]$. This result can be seen in Figure 6a. Equation (5) indicates that Q_1 and Q_3 are the first and third quartiles of the sequence $[X_{t_i} : X_{t_j}]$. It describes the scaled values for the data within the interquartile range, as shown in Figure 5b.

Equation (6) shows the magnitude warping method. This result can be seen in Figure 6b.

$\mathbf{w} = [w_1, w_2, \dots, w_T]$ represents the random warping curve generated from a normal distribution. $w_t \sim \mathcal{N}(1, \sigma^2)$ is each w_t sampled from a normal distribution with a mean of 1 and variance σ^2 . $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_t]$ represents the warped time series data. $\tilde{x}_t = x_t \cdot w_t$, each element x_t is the product of x_t and w_t .

Magnitude warping is a technique that smoothly transfers the magnitudes of time series data while preserving its temporal structure. As in Equation (6), we generate a random warping curve using the `np.random.normal` function to sample values from a normal distribution with a mean of 1.0 and a standard deviation of `sigma`. We then use `interp1d` to interpolate the generated warping curve for each time step of the time series data. This interpolated warping curve is applied to the original time series data by multiplying each element of the original data by the interpolated curve, thereby generating the warped time series data.

Flip, peak noise injection, and random constant value replacement are techniques referred to in AnomalyBERT [23].

In AnomalyBERT, the degradation scheme's "soft replacement" was expanded to include "external interval replacement." While "uniform replacement" might more accurately describe adding random noise from a uniform distribution, the original term was retained for ease of understanding. The term "peak noise" was also used as is.

Algorithm 1 shows the training process using the synthetic outlier generation module. The data are transformed using one of the following methods: flip, global scaling, local scaling, magnitude warping, arithmetic mean, geometric mean, and median.

Algorithm 1 Model training using synthetic outlier generation.

```

1: Load training data, replacing data, and test data
2: Initialize the Transformer backbone model
3: for iteration in the training loop do
4:   The data are transformed using one of the following methods: global scaling, local
   scaling, magnitude warping, arithmetic mean, geometric mean, median, and flip
5:   for iterate over batch to create anomaly intervals do
6:     Depending on the replacing type, apply appropriate replacing method
7:     if replacing method is external interval replacing then
8:       The data are transformed using one of the following methods: global scaling,
       local scaling, magnitude warping, arithmetic mean, geometric mean, median, flip
9:     else if replacing method is uniform replacement then
10:      Transform data using a random value
11:    else if replacing method is peak noising then
12:      Transform data using a random value
13:    end if
14:  end for
15:  Stack processed data into  $z$  and obtain model predictions  $y$ 
16:  Compute bce loss
17:  Estimate and evaluate model performance on test data
18: end for

```

Training

For our Transformer backbone (see AnomalyBERT [23]) utilizes binary cross-entropy loss for an input window with a replaced interval and the predicted anomaly scores.

The model was trained to classify all data points in a window as either normal or anomalous. At each training step, a synthetic outlier was introduced within the original window, replacing data with values of random length and at a random starting point within the window. The columns were modified at a 0.3 ratio, with the remainder left unchanged. There was a 50% chance of applying external interval replacement, a 15% chance of uniform replacement,

and a 15% chance of peak noise. Length adjustment was not applied in AnomalyBERT, as it decreased performance on some datasets.

3. Experiments and Results

3.1. Datasets

We used SWaT [27], WADI [28], SMAP [29], MSL [29], SMD [30], and IMOD datasets, as seen in Table 1.

SWaT (Secure Water Treatment): SWaT is a test bed dataset designed to simulate a secure water treatment system. It includes data on various stages of water treatment processes and is used for research in cybersecurity and anomaly detection in critical infrastructure. It has 51 features.

WADI (water distribution testbed): WADI is a water distribution system that provides data on the distribution and management of water resources. It is used to study anomalies and cybersecurity issues in water infrastructure. The dataset comprises 123 features.

SMAP (Soil Moisture Active Passive): The SMAP dataset consists of satellite data on soil moisture and freeze-thaw states from NASA. It is utilized for monitoring agricultural droughts, improving weather forecasts, and understanding climate dynamics. It includes 25 features.

MSL (Mars Science Laboratory): The MSL dataset contains telemetry data from the Mars Science Laboratory's Curiosity rover. It includes various measurements related to the rover's environment and operations on Mars, which are useful for fault detection and diagnostics. This dataset has 55 features.

SMD (Server Machine Dataset): The SMD originates from server machines in a data center and includes metrics such as CPU usage, memory, and network traffic. It is commonly used for studying anomaly detection and predictive maintenance in IT systems. It has 38 features.

IMOD (Industrial Maritime Operational IoT Data): IMOD is collected from engines and associated machinery on ships in actual operation. It includes data related to the internal components of the engine, such as cylinder and exhaust gas temperatures, coolant temperatures, cylinder pressures, oil pressures, engine fuel flow rates, fuel tank levels, and vibration and noise levels. Additionally, it encompasses data on pressure and rotational speed (RPM). A total of 31 key features were extracted and used for data analysis. Figure 3a shows key data for IMOD. The operating pressure of the cylinder (Pmax) refers to the maximum pressure reached within an engine cylinder during the combustion process, a crucial parameter indicating the peak pressure experienced by the cylinder, which directly impacts engine performance and efficiency. The cylinder exhaust gas outlet temperature measures the temperature of the exhaust gases as they exit the engine cylinder. Abnormal exhaust gas temperatures can signal issues such as incomplete combustion, overloading, or cooling system problems. Bearing Temperature refers to the temperature of the bearings in the engine. Bearings support and guide moving parts, and their temperature can indicate the health of the lubrication system and the bearing itself. A high bearing temperature can be a sign of insufficient lubrication, misalignment, or excessive load. Power measures the engine's output, with units in kilowatts (kW) or horsepower (HP). It reflects the engine's ability to convert fuel into mechanical energy. Load refers to the demand placed on the engine, measured as a percentage of the engine's maximum capacity.

Table 1. Datasets [23].

Dataset	Train	Test	Anomaly % in Test	Features
SWaT (2017)	495,000	449,919	12.13%	51
WADI (2017)	784,537	172,801	5.77%	123
MSL (2018)	58,317	73,729	10.53%	55
SMAP (2018)	153,183	427,617	12.79%	25
SMD (2019)	25,300 *	25,301 *	4.16%	38
IMOD (2023)	138,895	15,438	0.03%	31

* Note: Train and test lengths for SMD dataset are approximate. IMOD: Industrial Maritime Operational Dataset.

3.2. Training Setting

We used an NVIDIA GeForce RTX 3090 GPU for model training.

During the evaluation, the trained model processed the test data windows to predict anomaly scores for each data point. Using a sliding-window strategy, scores were averaged across overlapping intervals. Data points with scores above a certain threshold are classified as anomalies [23]. We applied the F1 score (F1) over the ground truth label and anomaly prediction for evaluation. We computed TP (true positive), FP (false positive), and FN (false negative) to obtain the F1 as $2TP/(2TP + FP + FN)$. The model was designed to classify the data as anomalous if any anomalous data were included within the window.

The model architecture uses AnomalyBERT [23] as the backbone, with a Transformer encoder having an embedding dimension of 512, a Transformer body with six layers, eight attention heads, and two linear layers. During training, the batch size was 8, and the maximum iterations were set to 150,000. Input windows were randomly selected from the training data and synthetic outliers were generated. Three types of outliers were applied in ratios of 6.25:1.875:1.875 for external interval replacement, uniform replacement, and peak noise, respectively. For external interval replacement, the model was trained with one of several techniques, such as global scaling, local scaling, magnitude warping, arithmetic mean, geometric mean, median, and flip.

As shown in Table 2, different patch sizes were applied to each dataset, ranging from 2 to 14. For the IMOD dataset, a patch size of 4 was chosen, referencing similar benchmark datasets. Window sizes varied from 1024 to 7168, with the IMOD dataset using a size of 2048. To prevent a decline in model performance, the ratio of outliers was kept below a certain threshold. Specifically, for the IMOD dataset, the outlier ratio was controlled to not exceed 15% of the total data.

The AdamW optimizer was used with a learning rate of 1×10^{-4} , with a warm-up of 10% and a decay of the cosine learning rate during the training phase.

All seven detailed methodologies for external interval replacement (arithmetic mean, geometric mean, median, global scale, local scale, magnitude warping, flip) were individually trained, and the model with the highest F1 score was used, as shown in Figure 3b.

Table 2. Settings for each dataset [23].

Dataset	WADI	SWaT	MSL	SMAP	SMD	IMOD
Patch size	8	14	2	4	4	4
Window size	4096	7168	1024	2048	2048	2048
Max length % of outlier	15%	20%	20%	15%	20%	15%

3.3. Experimental Results

As shown in Table 3, our approach outperformed AnomalyBERT in all six datasets. We were unable to replicate the results of the original paper on the WADI and SMD datasets. However, in our experimental setup, under the same conditions, our approach showed better performance compared to the original method. As shown in Figure 7, it can be visually confirmed through the bar graph that our approach demonstrates superior performance across all six datasets.

Table 3. F1 score compared with SOTA. Bold values represent the highest F1 score achieved across all models.

Type	WADI	SWaT	MSL	SMAP	SMD	IMOD
AnomalyBERT (2023) [23]	0.5	0.81347	0.302	0.457	0.25056	0.375
CARLA (2025) [31]	0.2953	0.72	0.5227	0.5292	0.5114	0.36750
Ours	0.52543	0.85982	0.37356	0.55283	0.26332	0.54545

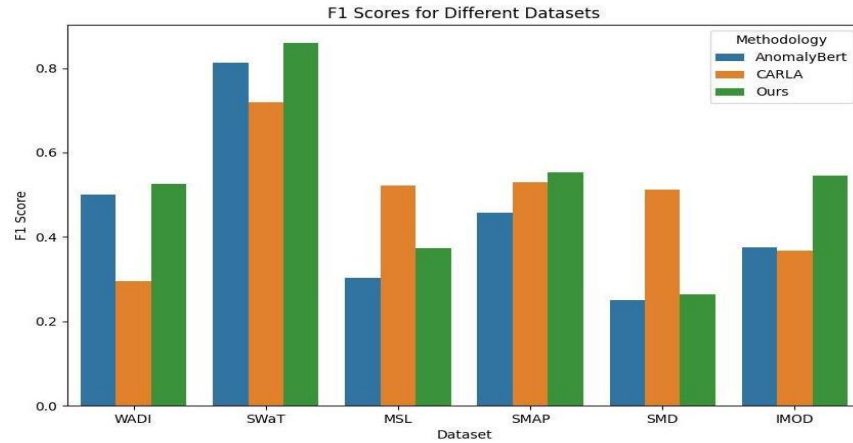


Figure 7. F1 score compared with SOTA. The blue bar represents AnomalyBERT, the orange bar represents CARLA, and the green bar represents the F1 score.

Table 4 shows the F1 scores for external interval replacement using various outlier generation methods. Flip was the default method for external interval replacement in AnomalyBERT [23]. The performance with different methods, including arithmetic mean, geometric mean, median, global scale, local scale, and magnitude warping, is summarized in the table.

For the WADI dataset, the global scaling method showed the best performance, resulting in a 5.08% improvement. In the SWaT dataset, local scaling improved performance by 5.69%. Magnitude warping was most effective for the MSL dataset, enhancing performance by 23.69%. The arithmetic mean improved SMAP by 18.38%, while the median was best for SMD with a 5.09% improvement. In the IMOD dataset, the F1 score increased from 0.375 with Flip to 0.54545 with global scaling, marking an improvement of 45.43%.

As seen in Figure 8, the most effective methods vary between different datasets.

Datasets with lower frequency data, like WADI, might benefit more from global scaling adjustments due to fewer fluctuations, whereas datasets with more frequent variations, such as SWaT, might see better performance with local scaling methods.

As seen in Table 5, a sensitivity analysis of the global scale was conducted. The global scale was adjusted using values ranging from the mean – std to mean + std, classified as min, 1Q, 2Q, 3Q, and max, and recorded as global scales 0, 1, 2, 3, and 4, respectively.

Table 4. F1 score improvement achieved through external interval replacement. Bold values represent the highest F1 score achieved across all models.

Type	WADI	SWaT	MSL	SMAP	SMD	IMOD
Flip	0.5	0.81347	0.302	0.457	0.25056	0.375
Arithmetic mean	0.50032	0.70690	0.24517	0.55283	0.23654	0.41667
Geometric mean	0.53	0.78232	0.25792	0.53213	0.21219	0.30303
Median	0.49199	0.76447	0.31979	0.47977	0.26332	0.41667
Global scale	0.52543	0.81739	0.33398	0.54104	0.25622	0.54545
Local scale	0.48747	0.85982	0.28279	0.51129	0.23867	0.3125
Magnitude warping	0.51117	0.81489	0.37356	0.53044	0.2366	0.3125
Random	0.47269	0.78627	0.28006	0.49883	0.22996	0.30303
Improvement	5.08%	5.69%	23.69%	18.38%	5.09%	45.43%

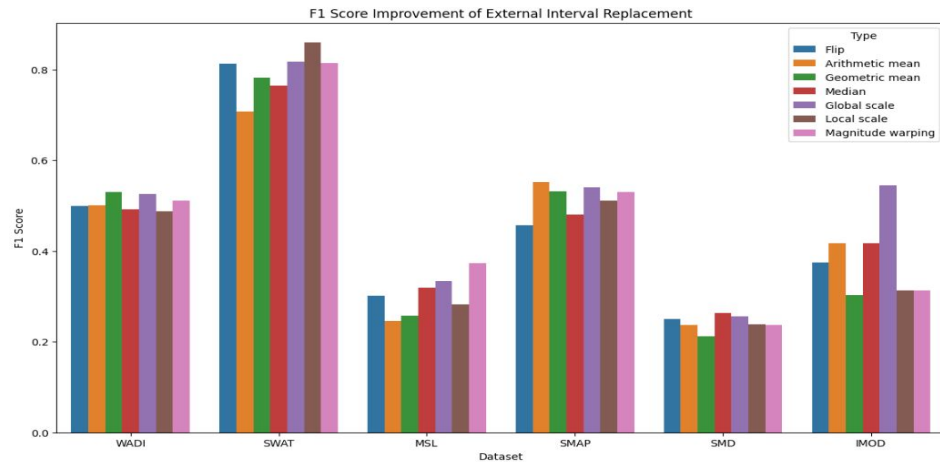


Figure 8. F1 score for external interval replacement. The blue bar represents the Flip method, the orange bar denotes the arithmetic mean method, the green bar represents the geometric mean method, and the red bar represents the median method. The purple bar denotes the global scale method. The brown bar represents the local scale method. The Pink bar represents the magnitude warping method.

Table 5. Performance metrics across 6 datasets with global scaling sensitivity. Global scale 0 sets the measurement to the window’s mean minus the standard deviation. Global scale 1 corresponds to the first quartile (1Q) within the window. Global scale 2 represents the median (2Q), global scale 3 aligns with the third quartile (3Q), and global scale 4 adjusts to the window’s mean plus the standard deviation. Bold values represent the highest F1 score achieved across all models.

Type	WADI	SWaT	MSL	SMAP	SMD	IMOD
Global scale 0	0.50787	0.76447	0.29306	0.54104	0.21219	0.48485
Global scale 1	0.51342	0.81023	0.29735	0.52026	0.22672	0.5
Global scale 2	0.52543	0.79751	0.33398	0.51284	0.23316	0.48485
Global scale 3	0.49171	0.75329	0.32729	0.48407	0.25622	0.48485
Global scale 4	0.51594	0.81739	0.32914	0.50494	0.23814	0.54545

The WADI dataset showed the best performance with a global scale of 2 (0.52543); SWaT had a global scale of 4 (0.81739), and MSL had a global scale of 0 (0.54104). SMD performed best with a global scale of 3 (0.25622), as well as IMOD, with a global scale of 4 (0.54545). WADI, a low-frequency dataset, showed improved performance at the median. The sensitivity analysis did not show a clear pattern for SWaT, SMAP, and MSL. Apart from SMAP, higher global scale values (2Q, 3Q, max) generally improved performance, as seen in Figure 9.

Figure 10 shows the graph of the loss decreasing as our globally set model trains. Initially, the loss starts near a value of 1, but it gradually decreases over the iterations. After 20,000 iterations, the loss drops below 0.0013115085, and after 150,000 iterations, the loss decreases to 0.0000551652, indicating that the training is progressing well.

Figure 11 shows a graph comparing the original data with the matched anomaly score. The original data, representing Cylinder7 Pmax (peak maximum pressure), usually maintains a value between 50 and 60 bar. However, a sudden drop in pressure is observed around the 10,000 to 11,000 index, signifying an anomaly. During this time, the anomaly score from our model spikes close to 1.0. Typically, the Pmax value ranges from approximately 80 to 200 bar for a modern marine diesel engine. The model effectively detected the anomaly, as indicated by the significant drop in the Pmax value.

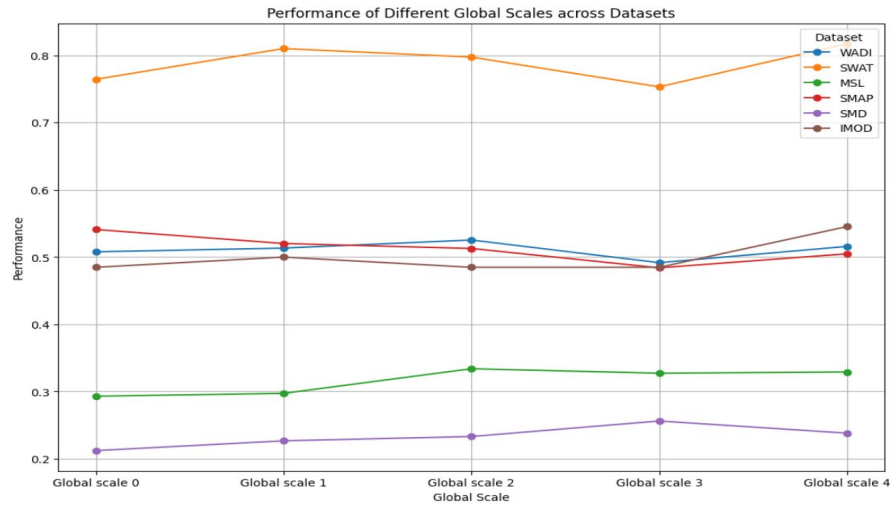


Figure 9. Performances of different global scales for different datasets. WADI (blue), SWaT (orange), MSL (green), SMAP (red), SMD (purple), and IMOD (brown).

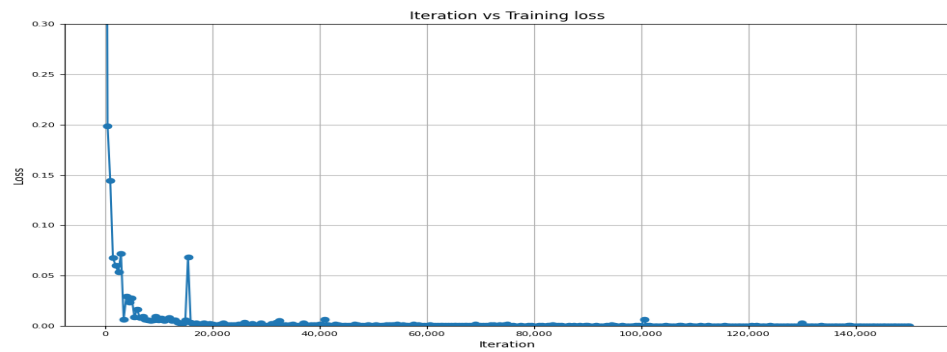


Figure 10. Loss with global scales for the IMOD dataset.

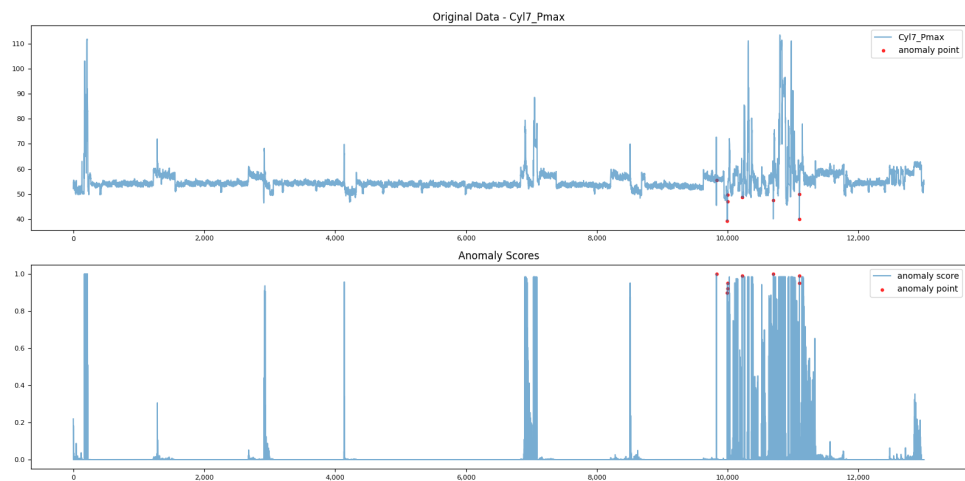


Figure 11. Anomaly detection results for IMOD. The red point denotes the test dataset. The top part shows the original data for Cylinder7 Pmax, while the bottom part displays the anomaly score results for the corresponding indices.

The original data in Figure 12 denote the bearing temperature, which typically remains around 75 degrees Celsius. However, an abnormal pattern emerges when the temperature suddenly drops to 45 degrees before rising again. The normal temperature range for general bearings is approximately 60 to 80 degrees Celsius. Deviations from this range may indicate issues such as lubrication problems, overload, or misalignment.

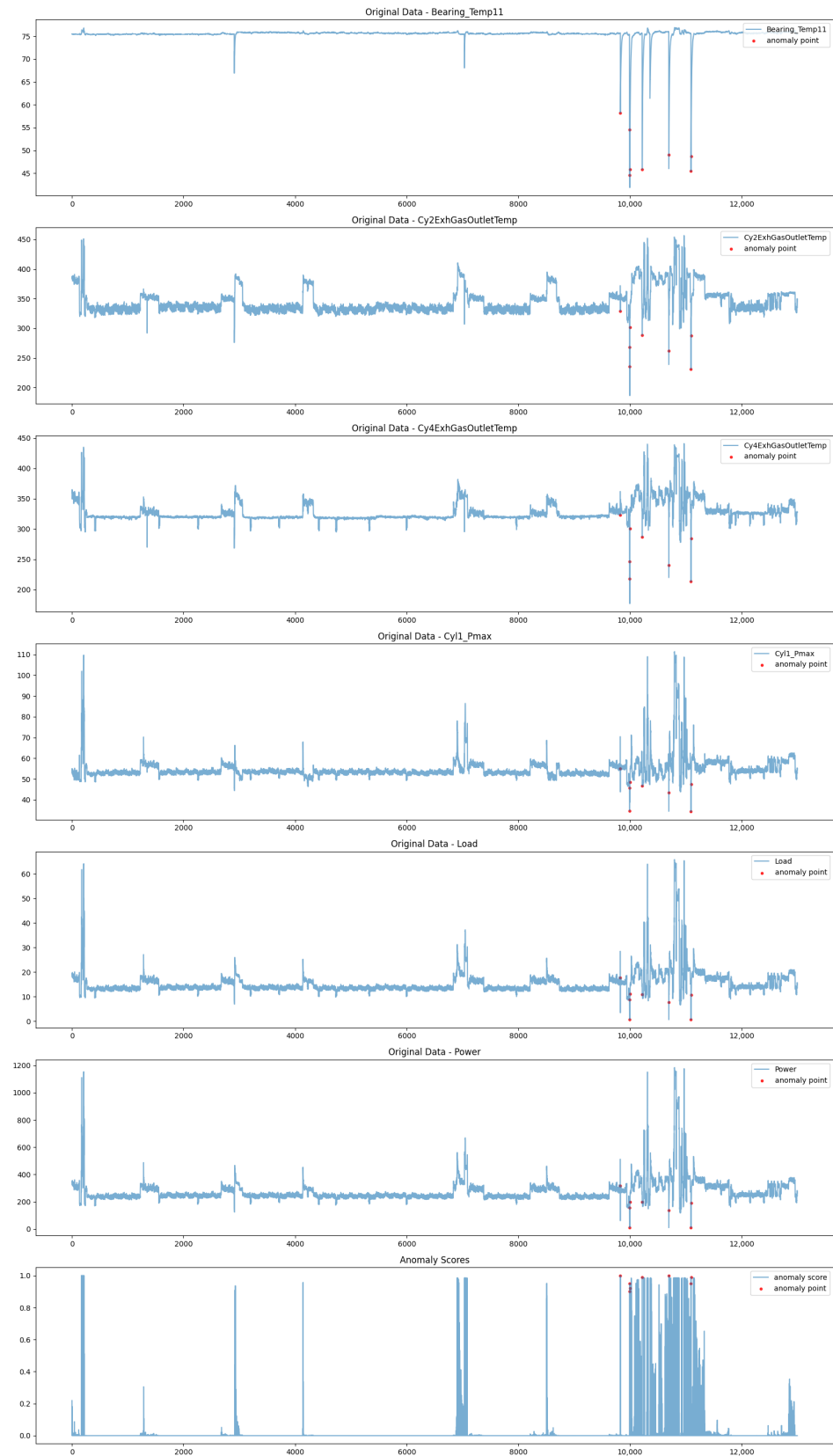


Figure 12. Anomaly detection results for IMOD. The red point is the label for the test dataset. The top part shows the original data for the bearing temperature, cylinder exhaust gas outlet temperature, cylinder Pmax, engine load, and power, while the bottom part displays the anomaly score results for the corresponding indices.

The second and third original datasets in Figure 12 represent the cylinder exhaust gas outlet temperature, which typically ranges between 300 and 350 degrees Celsius, but the temperature suddenly dropped to around 200 degrees and then spiked to over 400 degrees, indicating an abnormal pattern. For diesel engines, excessively high exhaust gas temperatures can indicate combustion issues, overload, or fuel quality problems. Conversely, excessively low exhaust gas temperatures can indicate incomplete combustion, low engine load, or cooling system problems.

The fourth original dataset in Figure 12 represents the cylinder Pmax. The normal range is maintained between 50 and 60, but it suddenly dropped below 40 and then spiked to around 110, showing an abnormal pattern.

The fifth original dataset in Figure 12 represents the engine load. The normal range is maintained between 10% and 20%, but it suddenly dropped to around 0% and then spiked to over 60%, showing an abnormal pattern. If the engine load is too low, the engine operates inefficiently, and if it is too high, there is an increased risk of overheating and damage.

The sixth original dataset in Figure 12 represents engine power. Engine power represents the amount of output produced by the ship's engine and is measured in kilowatts (kW). Normally, it maintains values between 200 and 400 kW, but it suddenly dropped below 100 and then spiked to 1200, showing an abnormal pattern. Engine power is directly related to the propulsion of the ship and is a key factor in determining the ship's speed and maneuverability.

The seventh graph in Figure 12 shows the anomaly score detected by our model. At the points corresponding to the abnormal patterns observed in the bearing temperature, cylinder exhaust gas outlet temperature, cylinder Pmax, engine load, and engine power, the anomaly score was significantly predicted.

4. Discussion

Q1: Did introducing regularity into the data improve the model's performance?

A1: Yes, introducing regularity into the data improved the model's performance. The experiment compared two methods: random replacement and regular replacement. We tested the performance by replacing the extracted data intervals with completely random values versus replacing them with values that followed a regular pattern, such as global values. As shown in Table 4, the random replacement method yielded the lowest performance across all six datasets, while the regular replacement method showed better results. This indicates that introducing regularity to the data is a significant factor in enhancing performance.

5. Conclusions

Through our outlier synthetic framework, we achieved an average improvement of 17.23% in the F1 score for anomaly detection performance compared to the state-of-the-art across five benchmark time series datasets and our own collected maritime operational dataset. Specifically, the anomaly detection performance on our maritime operational dataset improved from an F1 score of 0.375 with the SOTA method to 0.54545 using our approach, representing a 45.43% increase.

The model leverages a Transformer backbone and enhances performance by generating three types of synthetic outliers: external interval replacement, uniform replacement, and peak noise. The external interval replacement method involves degrading data using arithmetic mean, geometric mean, and flip methods, or augmenting data with magnitude warping, global scaling, and local scaling.

Each method showed varying effectiveness, depending on the characteristics of the dataset. For example, global scaling performed best on low-frequency data such as the WADI dataset, as well as on our maritime operational dataset.

While Transformer models are becoming increasingly complex and powerful, our study demonstrates that simpler data manipulation techniques, like synthetic outlier generation, can provide cost-effective solutions for anomaly detection. This makes our contribu-

tion significant, showing the potential for efficient and effective anomaly detection using relatively straightforward approaches.

Although we improved performance through the synthetic outlier framework, we could not precisely explain why this approach impacts performance. Future research will continue to explore how different outlier generation methods affect performance based on the characteristics of the data.

Author Contributions: Conceptualization, H.K. and I.J.; methodology, H.K.; software, H.K.; validation, H.K. and I.J.; formal analysis, H.K.; investigation, H.K.; resources, H.K.; data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, H.K. and I.J.; visualization, H.K.; supervision, I.J.; project administration, I.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset presented in this article is not readily available because the data belong to a private company.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Launch of the Review of Maritime Transport 2024.
2. Xu, G.; Shi, Y.; Sun, X.; Shen, W. Internet of things in marine environment monitoring: A review. *Sensors* **2019**, *19*, 1711. [[CrossRef](#)] [[PubMed](#)]
3. Androjna, A.; Brcko, T.; Pavic, I.; Greidanus, H. Assessing cyber challenges of maritime navigation. *J. Mar. Sci. Eng.* **2020**, *8*, 776. [[CrossRef](#)]
4. Lai, K.; Zha, D.; Wang, G.; Xu, J.; Zhao, Y.; Kumar, D.; Chen, Y.; Zumkhawaka, P.; Wan, M.; Martinez, D.; et al. TODS: An Automated Time Series Outlier Detection System. *arXiv* **2020**, arXiv:2009.09822. [[CrossRef](#)]
5. Wei, Y.; Jang-Jaccard, J.; Xu, W.; Sabrina, F.; Camtepe, S.; Boulic, M. LSTM-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sensors J.* **2023**, *23*, 3787–3800. [[CrossRef](#)]
6. Saha, S.; Sarkar, J.; Dhavala, S.; Sarkar, S.; Mota, P. Quantile LSTM: A Robust LSTM for Anomaly Detection In Time Series Data. *arXiv* **2023**, arXiv:2302.08712. [[CrossRef](#)]
7. Zhao, Z.; Xu, C.; Li, B. A LSTM-based anomaly detection model for log analysis. *J. Signal Process. Syst.* **2021**, *93*, 745–751. [[CrossRef](#)]
8. Wen, T.; Keyes, R. Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv* **2019**, arXiv:1905.13628. [[CrossRef](#)]
9. Guo, W.; Liu, X.; Xiang, L. Membrane system-based improved neural networks for time-series anomaly detection. *Processes* **2020**, *8*, 1168. [[CrossRef](#)]
10. Choi, T.; Lee, D.; Jung, Y.; Choi, H.J. Multivariate time-series anomaly detection using SeqVAE-CNN hybrid model. In Proceedings of the 2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 12–15 January 2022; pp. 250–253. [[CrossRef](#)]
11. Minhas, M.S.; Zelek, J. Semi-supervised Anomaly Detection using AutoEncoders. *arXiv* **2020**, arXiv:2001.03674. [[CrossRef](#)]
12. Zhou, C.; Paffenroth, R.C. Anomaly Detection with Robust Deep Autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2017; KDD '17, pp. 665–674. [[CrossRef](#)]
13. Niu, Z.; Yu, K.; Wu, X. LSTM-Based VAE-GAN for Time-Series Anomaly Detection. *Sensors* **2020**, *20*, 3738. [[CrossRef](#)] [[PubMed](#)]
14. Li, L.; Yan, J.; Wang, H.; Jin, Y. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1177–1191. [[CrossRef](#)] [[PubMed](#)]
15. Li, D.; Chen, D.; Jin, B.; Shi, L.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; pp. 703–716. [[CrossRef](#)]
16. Geiger, A.; Liu, D.; Alnegheimish, S.; Cuesta-Infante, A.; Veeramachaneni, K. Tadgan: Time series anomaly detection using generative adversarial networks. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 33–43. [[CrossRef](#)]
17. Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; Xu, B. TS2Vec: Towards Universal Representation of Time Series. *arxiv* **2022**, arXiv:cs.LG/2106.10466. [[CrossRef](#)]
18. Le Guennec, A.; Malinowski, S.; Tavenard, R. Data augmentation for time series classification using convolutional neural networks. In Proceedings of the ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Grenoble, France, 19–23 September 2016.

19. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv* **2018**, arXiv:1808.02455. [[CrossRef](#)]
20. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time Series Data Augmentation for Deep Learning: A Survey. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, Virtual, 19–26 August 2021; p. IJCAI-2021. [[CrossRef](#)]
21. Um, T.T.; Pfister, F.M.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; Kulić, D. Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, 13–17 November 2017; pp. 216–220. [[CrossRef](#)]
22. Fan, H.; Zhang, F.; Wang, R.; Huang, X.; Li, Z. Semi-Supervised Time Series Classification by Temporal Relation Prediction. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 3545–3549. [[CrossRef](#)]
23. Jeong, Y.; Yang, E.; Ryu, J.H.; Park, I.; Kang, M. AnomalyBERT: Self-Supervised Transformer for Time Series Anomaly Detection using Data Degradation Scheme. *arXiv* **2023**, arXiv:cs.LG/2305.04468. [[CrossRef](#)]
24. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *arXiv* **2021**, arXiv:cs.LG/2012.07436. [[CrossRef](#)]
25. Kang, H.; Kang, P. Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism. *Knowl. Based Syst.* **2024**, *290*, 111507. [[CrossRef](#)]
26. Darban, Z.Z.; Yang, Y.; Webb, G.I.; Aggarwal, C.C.; Wen, Q.; Salehi, M. DACAD: Domain Adaptation Contrastive Learning for Anomaly Detection in Multivariate Time Series. *arXiv* **2024**, arXiv:cs.LG/2404.11269. [[CrossRef](#)]
27. Goh, J.; Adepun, S.; Junejo, K.; Mathur, A. *A Dataset to Support Research in the Design of Secure Water Treatment Systems*; Springer: Berlin/Heidelberg, Germany, 2016. [[CrossRef](#)]
28. Ahmed, C.M.; Palleti, V.R.; Mathur, A.P. WADI: A water distribution testbed for research in the design of secure cyber physical systems. In Proceedings of the Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, Pittsburgh, PA, USA, 21 April 2017. [[CrossRef](#)]
29. Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Söderström, T. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. *arXiv* **2018**, arXiv:1802.04431. [[CrossRef](#)]
30. 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, Anchorage, AK, USA, 4–8 August 2019. [[CrossRef](#)]
31. Darban, Z.Z.; Webb, G.I.; Pan, S.; Aggarwal, C.C.; Salehi, M. CARLA: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognit.* **2025**, *157*, 110874. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.