Mixer-Transformer: Adaptive Anomaly Detection with Multivariate Time Series

Xing Fang[a], Yuanfang Chen[a,b,*], Zakirul Alam Bhuiyan[c], Xiajun He[a], Guangxu Bian[a], Noel Crespi[d], Xiaoyuan Jing[e,f]

[a] *Hangzhou Dianzi University, 310018, Hangzhou, China*
[b] *The State Key Laboratory of Blockchain and Data Security, Zhejiang University, 310027, Hangzhou, China*
[c] *Fordham University, 10458, New York, USA*
[d] *Telecom SudParis, Institut Polytechnique de Paris, 91120, Palaiseau, France*
[e] *Wuhan University, 430072, Wuhan, China*
[f] *Guangdong University of Petrochemical Technolog, 528000, Maoming, China*

Abstract

Anomaly detection is crucial for maintaining the stability and security of systems. However, anomaly detection systems often generate numerous false positives or irrelevant alerts, which obscure genuine security threats. To both reduce false positives in time series detection and accurately identify the source of anomalies, leveraging artificial intelligence techniques has emerged as a promising solution. These techniques can analyze strong temporal correlations and dynamic variations across different data frames. Existing detection methods face two primary challenges leading to false positives or negatives: (i) detecting anomalies in multivariate time series requires accounting for both temporal dependencies and complex interactions between variables; and (ii) traditional fixed-threshold approaches often struggle to adapt to dynamic environments. To address these issues, this paper proposes an anomaly detection method based on the Mixer-Transformer architecture. By combining the Mixer model with the Anomaly Transformer, the proposed method effectively captures global dependencies by alternately modeling interactions along both the channel and time dimensions, thereby enhancing its ability to extract complex spatiotemporal features. Additionally, an adaptive threshold update mechanism is employed to dynamically adjust the anomaly detection criteria in response to data fluctuations. The F1 scores on three real-world datasets—SMAP, MSL, and PSM—are 97.49%, 95.18%, and 98.20%, respectively. These results demonstrate that the proposed method outperforms existing technologies in reducing false positives and enhancing the detection accuracy of multivariate time series anomaly detection.

Keywords: Anomaly Detection, Time Series, Mixer, Adaptive Threshold.

*Corresponding author
Email address: yuanfang.chen.tina@gmail.com (Yuanfang Chen)

1. Introduction

With the rapid advancement of artificial intelligence (AI) and the Internet of things (IoT), critical systems—such as industrial production, intelligent transportation, financial transactions, and medical monitoring—are generating vast amounts of time-series data [1, 2, 3]. These data not only reflect system's operational status but also contain valuable insights into equipment performance and potential faults. Anomaly detection in time-series data is essential for ensuring the secure operation of these systems, optimizing performance, and preventing losses. It helps identify hidden fault patterns, detect sudden abnormal events, and recognize potential risk signals, enabling timely intervention before issues escalate [4]. As a result, anomaly detection plays a vital role in modern industrial and intelligent systems, making it a core technology for data-driven security monitoring.

Time-series anomaly detection continues to face several challenges. First, anomalies are typically rare and diverse [5, 6], often hidden within large volumes of normal data. Anomaly detection systems are commonly burdened by numerous false positives or irrelevant alarms [7], which can obscure genuine security threats, waste resources, and reduce detection efficiency, all detrimental aspects given that models need to possess robust anomaly recognition capabilities. Second, time-series data often exhibit non-linear, non-stationary characteristics and complex inter-variable correlations [8, 9], presenting significant modeling challenges for traditional detection methods. Furthermore, many existing approaches rely on fixed thresholds to identify anomalies, which have notable limitations. Fixed thresholds fail to account for the dynamic variations and local fluctuations inherent in time-series data, making them poorly suited to handle the complex and variable nature of anomaly distributions. As a result, detection performance may degrade. These challenges highlight the need for more flexible and intelligent techniques to effectively address the complexities of time-series anomaly detection.

In response to the challenges outlined above, recent research has proposed various AI-based methods, such as modeling global temporal dependencies using self-attention mechanisms and combining generative models to capture the dynamic features of time-series data [10]. However, as shown in Figure 1, these methods have not fully explored cross-dimensional interactions within multivariate time-series data and are susceptible to interference from noise in scenarios with sparse anomalies. Additionally, the use of fixed-threshold approaches for anomaly detection across different application scenarios presents notable limitations.

To address the aforementioned issues, we propose a time series anomaly detection method that integrates multi-feature fusion and adaptive threshold adjustment. This approach combines the Mixer architecture with the Anomaly Transformer model to simultaneously model the interdependencies between the time dimension and feature channels. Specifically, the Mixer architecture excels

at capturing both local and global dependencies in time series data by employing a multi-channel alternating modeling mechanism. Meanwhile, the Anomaly Transformer model enhances the capture of long-range dependencies through its self-attention mechanism, boosting the model's ability to extract complex dynamic spatiotemporal features. To enhance adaptability, an adaptive threshold update mechanism is introduced, which dynamically adjusts anomaly detection criteria based on shifts in the model's predictions and historical data. The threshold is automatically modified in response to changes in data distribution, ensuring the consistency and accuracy of detection criteria and data features. Consequently, the proposed model effectively handles unstable or changing environments, providing more flexible and accurate anomaly detection capabilities. The main contributions of our work are as follows:

- The proposed Mixer-Transformer framework introduces hybrid feature fusion and alternately models the interaction between channel and time dimension information, thereby improving performance in handling complex time series data, particularly with long time spans and multi-dimensional data;
- An adaptive threshold update mechanism is designed to automatically adjust the threshold according to dynamic changes in the data, enabling the model to handle different anomaly patterns with greater flexibility and adaptability;
- The effectiveness of the proposed method is verified on three real-world datasets through ablation experiments, which comprehensively analyze the contribution of each component to overall performance and confirm the key role of the Mixer-Transformer architecture and adaptive threshold mechanism in enhancing anomaly detection capabilities.

The rest of this paper is organized as follows: Section 2 reviews related works. Section 3 outlines the research questions. Section 4 presents the proposed Mixer-Transformer framework. Section 5 details the algorithms and execution steps. Section 6 discusses the comparison and ablation experiments with the baseline. and Section 7 concludes the paper and outlines future research directions.

2. Related Work

In the field of anomaly detection, researchers have proposed various AI-based methods to address anomalies in different types of data. Early research focused primarily on statistical and traditional machine learning models. However, with advancements in technology, deep learning and other sophisticated models have gradually become mainstream, providing more effective solutions to the complexity and nonlinearity of time-series data. The goal of anomaly detection is to identify behaviors or data points that deviate from normal patterns, a process that requires a deep understanding of the data's features and accurate

modeling. As data dimensions and complexity increase, the limitations of traditional methods have become more evident, driving the continuous development of new approaches based on deep learning, graph neural networks, and adaptive thresholds.

Table 1: Comparison of Methods Based on Various Criteria (✓: Fully supported, △: Partially supported, ×: Not supported)

Category	**Method/Model**	**Temporal Dependency**	**Multivariate Interaction**	**Threshold Mechanism**	**Adaptability to Dynamics**
Statistical & ML	ARIMA, Isolation Forest [11, 12, 13, 14, 15]	△ (Linear, local)	×	× (Fixed)	×
Deep Learning Basics	LSTM, CNN, Transformer [16, 17, 18, 19, 20, 21, 22, 23, 24, 25]	✓ (Nonlinear, global)	△ (Implicit)	× (Fixed)	△ (Partial)
GNN + Adaptive Threshold	GNN-based models [19, 26, 27, 28, 29, 30]	✓ (Graph-based)	✓ (Explicit via graph)	✓ (Adaptive, e.g., EWMA)	✓
Proposed Method	Mixer-Transformer + Dynamic Threshold	✓ (Global+Local via Mixer-Transformer)	✓ (Explicit channel-time mixing)	✓ (Adaptive, dynamic updates)	✓

2.1. Anomaly Detection in Statistics and Machine Learning

Traditional methods for anomaly detection primarily rely on statistical and conventional machine learning models to capture the characteristics of time-series data. Statistical methods detect anomalies by analyzing the mean, variance, and autoregressive properties of time series (e.g., ARIMA models) [11, 12, 13]. However, these methods assume data stationarity, making it difficult to handle the complex patterns of non-stationary or nonlinear data encountered in real-world scenarios. Unsupervised learning methods, such as Isolation Forest, identify outliers by learning the distribution of normal data [14, 15] . While these methods address some of the limitations of statistical approaches, they struggle to model the dependencies in high-dimensional, multivariate data. Furthermore, fixed-threshold methods, which are commonly used for anomaly detection, fail to account for the dynamic changes and local patterns inherent in time-series data.

2.2. Anomaly Detection Based on Deep Learning

The rise of deep learning has introduced a new paradigm in time-series anomaly detection, significantly enhancing detection performance through nonlinear feature extraction and complex pattern modeling. Recurrent networks, such as LSTM, effectively capture temporal dependencies [16, 17, 18], while convolutional neural networks (CNNs) excel at extracting local features [19]. However, both methods struggle to handle long-range dependencies and multivariate interactions. In recent years, Transformer-based self-attention methods have shown exceptional performance in time-series modeling [20, 21, 22, 23]. For example, the Anomaly Transformer [20] detects anomalies by comparing

the differences between prior-associations and series-associations in time-series data. While the self-attention mechanism facilitates global modeling of relationships [24], existing methods still leave room for improvement, particularly in terms of modeling multivariate feature interactions and representing anomaly patterns[25].

2.3. Anomaly Detection in Multivariate Time-Series with Adaptive Thresholds

Anomaly detection in multivariate time-series data requires addressing both temporal dependencies and complex interactions between variables [19, 26, 27, 28]. Some studies have leveraged Graph Neural Networks (GNNs), to model the relationships between variables as a graph structure [29]. However, these approaches often suffer from high computational complexity and limited performance in scenarios with sparse anomalies. Additionally, traditional fixed-threshold anomaly detection methods lack the flexibility needed to effectively adapt to the dynamic patterns of time-series data. Recently, some studies have explored adaptive thresholding [30], dynamically adjusting detection criteria using techniques such as sliding windows and exponentially weighted moving averages, thereby enhancing the model's ability to adapt to complex anomaly patterns.

3. Problem Description

Anomalies in time series data are typically characterized by their rarity and weak association with the overall dataset. These outliers often manifest at specific, localized time points where their relationships are concentrated. This localized concentration forms a key distinguishing feature, known as 'Association Discrepancy,' which can be used to differentiate normal points from anomalies. Understanding this discrepancy is crucial for time series anomaly detection, as intuitively explained in a Bilibili video by blogger Mardinff, and illustrated in Figure 1. In real-world applications, however, data often exhibit complex cross-variable relationships and auxiliary features, which can significantly influence anomaly patterns. As such, anomaly detection methods must account not only for temporal dependencies but also for the combined impact of related variables.

The primary goal of this research is to develop an advanced time series anomaly detection method that effectively captures the subtle and localized nature of anomalies by considering both temporal dependencies and cross-variable relationships. The focus is on creating a flexible detection framework capable of adapting to varying data characteristics, particularly in environments where anomalies evolve over time.

Several challenges must be addressed in this study. First, the presence of multiple related variables complicates the task of isolating anomalies based solely on individual features, necessitating the careful integration of auxiliary features and cross-variable interactions into the anomaly detection process. Moreover, the dynamic nature of real-world data further complicates anomaly detection. A fixed-threshold approach, commonly used in traditional methods,

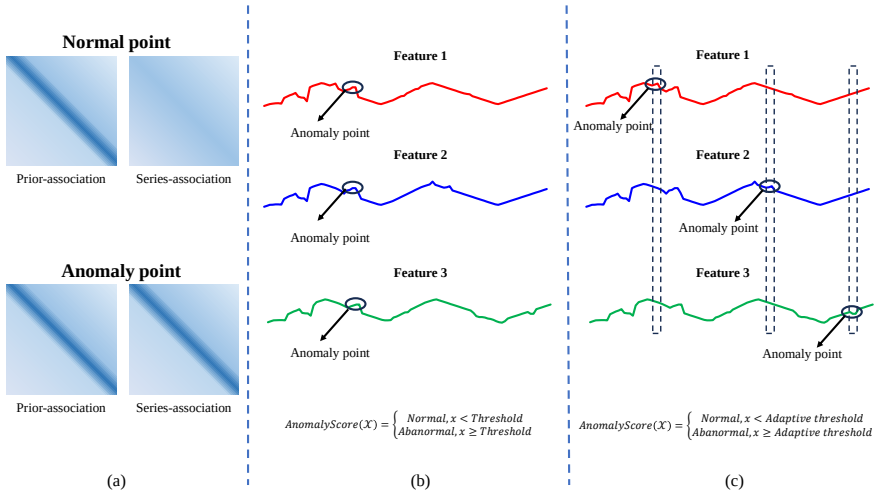

Figure 1: (a) By leveraging the differences between prior-association and series-association, anomalies can be effectively distinguished from normal data points.(b) Existing methods typically focus on univariate data or single-point observations, neglecting the contextual information in multivariate data and the complex dependencies between variables. Additionally, fixed-threshold methods often fail to account for the dynamic nature of time-series data, leading to limitations in their effectiveness.(c) The use of alternating modeling between channel and time dimension interactions effectively captures global dependencies and enhances the model's ability to extract complex dynamic spatiotemporal features. At the same time, an adaptive threshold update mechanism is employed to flexibly adjust the anomaly detection criteria based on dynamic changes in the data.

is insufficient as it fails to account for the changing characteristics of time series data, leading to suboptimal performance.

4. Building Proposed Mixer-Transformer Model

In this section, the Mixer-Transformer is introduced, incorporating the Mixer structure. By alternately modeling interactions between the channel and time dimensions, this approach effectively integrates both local and global feature information, providing a more comprehensive and abstract representation of time series data for the Anomaly Transformer. Additionally, an adaptive threshold is implemented during outlier detection to enhance the model's adaptability to complex anomaly patterns.

4.1. Preliminaries

Due to the rarity of anomalies, anomalous points are often weakly associated with an entire time series, with their associations primarily concentrated at adjacent time points. This concentration of associations at nearby points creates a distinguishable criterion for differentiating normal from anomalous points, referred to as association discrepancy. The discrepancy is quantified by comparing

the prior-association with the series-association at each time point, calculating the difference using symmetric Kullback-Leibler (KL) divergence.

A prior-association provides the model with an initial hypothesis of associations based on relationships between adjacent time points, reflecting the expected pattern of local associations within the time series. During the early stages of model training, this helps guide the model to focus on the relationships between adjacent time points, preventing blind exploration and accelerating convergence. Additionally, due to the rarity of anomalies, their associations are typically more concentrated around adjacent time points. This characteristic of the prior-association enables the model to become sensitive to anomaly patterns early in training, establishing a foundation for accurate anomaly detection in subsequent stages. The association weight of each time point relative to others is computed using a learnable Gaussian kernel, as shown in the following:

$$\mathrm{G}(|j-i|;\sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right) \tag{1}$$

where $i, j \in \{1, \ldots, N\}$ represent the i-th and j-th time points, respectively, and σ_i is the learnable scale parameter corresponding to the i-th time point. Based on the properties of the Gaussian distribution, the association weight is computed according to the relative distance $|j-i|$ between time points. The closer the distance, the larger the weight, reflecting the assumption that adjacent time points in the time-series are more strongly associated. This embodies the Adjacent-Concentration Inductive Bias, which posits that the association between adjacent time points is more concentrated.

Therefore, the prior-association PA can be represented as in Eq.(2):

$$PA = \mathrm{Softmax}\left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{|j-i|^2}{2\sigma_i^2}\right)\right] \tag{2}$$

The association weights of each row are normalized to create a discrete distribution. This approach adopts the normalization technique from the attention mechanism, which transforms the prior association strength between adjacent time points into a probabilistic form, effectively quantifying local dependencies.

Series-association refers to the relationship that is adaptively learned from the raw time-series data, enabling the model to capture dynamic dependencies within the time series and reflect the true degree of association between time points. Unlike prior-association, which is based on a predefined pattern, series-association is learned directly from the data itself. The calculation of series-association is performed through a standard self-attention mechanism, as shown in the following Eq.(3) and Eq.(4):

$$Q, K, V = X^{(l-1)}W_Q^l, X^{(l-1)}W_K^l, X^{(l-1)}W_V^l \tag{3}$$

$$SA = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_{\text{model}}}}\right) \tag{4}$$

where $X^{(l-1)}$ represents the output of the $(l-1)$-th layer, $l \in \{1, 2, \ldots, L\}$, and W_Q^l, W_K^l, W_V^l are the learnable parameter matrices. The dimensions of Q,K,V are $N \times d_{\text{model}}$,where N is the length of the time-series and d_{model} is the number of channels in the model's hidden state. QK^T represents the dot product of the query Q and the key K, resulting in an attention score matrix of dimensions $N \times N$.

In a multi-layer model, PA and SA represent the set of computation results for each layer and can be expressed as:

$$PA = \{PA^1, PA^2, \ldots, PA^L\} \tag{5}$$

$$SA = \{SA^1, SA^2, \ldots, SA^L\} \tag{6}$$

To more comprehensively measure the difference between two distributions and avoid biases caused by the order of the distributions, symmetric Kullback-Leibler divergence is used (considering both $KL(PA_i^l \parallel SA_i^l)$ and $KL(PA_i^l \parallel SA_i^l)$) to calculate the difference between prior-association and series-association. The KL divergence for the i-th time point at the l-th layer is defined as:

$$KL(PA_i^l \mid QA_i^l) = \sum_i p_i \log \frac{p_i}{s_i} \tag{7}$$

In a multi-layer model, to consider the association information learned at different layers, the association discrepancies from each layer are fused to obtain a more representative and stable association discrepancy metric. By averaging, the model can reduce the impact of a poorly learned or anomalous layer on the overall association discrepancy calculation, thereby more accurately reflecting the association characteristics of the time points in the time-series. The association discrepancy between PA and SA in a multi-layer model can be computed by the following:

$$\text{AssDis}(PA, SA; X) = \frac{1}{L} \sum_{l=1}^{L} \Big[KL(PA_i^l \parallel SA_i^l) + KL(SA_i^l \parallel PA_i^l) \Big] \tag{8}$$

The association difference quantifies the distributional discrepancy between the prior association and the serial association. A larger AssDis value indicates a greater deviation of the association pattern at the current time point from the typical local adjacent dependency characteristics, thereby reflecting the severity of the anomaly.

In summary, the anomaly score for each time point in the time-series can be defined as:

$$\text{AnomalyScore}(X) = \text{Softmax}\Big(- \text{AssDis}(PA, SA; X) \Big) \odot \Big[\|X_i - \hat{X}_i\|_2^2 \Big] \tag{9}$$

where $\hat{X}_i$ is the reconstructed result of the input time-series data X, $i \in \{1, 2, \ldots, N\}$, and $\odot$ is the element-wise multiplication.

4.2. Mixer-Transformer to Anomaly Transformer Model's Encoder

We incorporate the Mixer structure into the front end of the Anomaly Transformer model's encoder.The architecture of Mixer-Transformer is shown in Figure 2 . The Mixer consists of a series of Mixer Layers, each containing two fully connected sublayers: a Feature Mixing sublayer and a Time Mixing sublayer. The Feature Mixing sublayer focuses on mixing input features along the channel dimension, capturing interactions between different channels. The Time Mixing sublayer, on the other hand, mixes features along the time dimension, modeling the temporal dependencies between different time steps. The outputs of these two sublayers are combined through residual connections and then passed through Layer Normalization to ensure stability and training effectiveness, resulting in the final output of the layer.

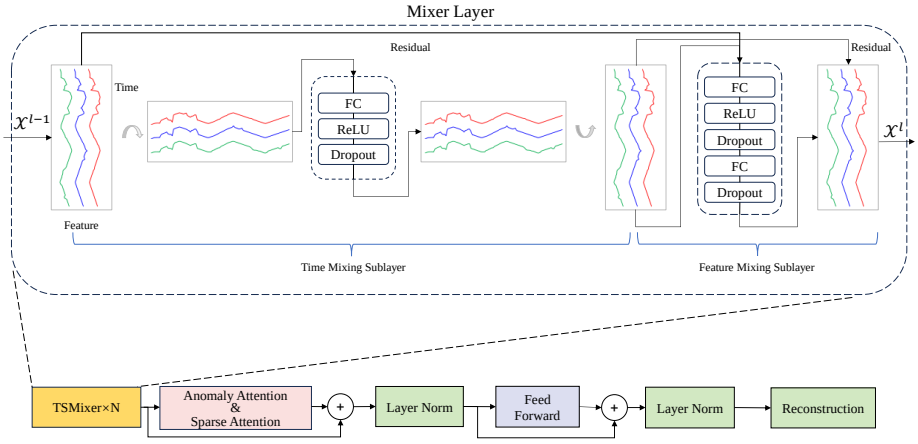

Figure 2: Architecture of Mixer-Transformer.

Specifically, for the l-th Mixer Layer, the input is $X^{(l-1)} \in \mathbb{R}^{B\times T\times C}$, where B is the batch size, T is the number of time steps, and C is the number of feature channels. The Time Mixing sublayer first reshapes the input into the form $X_t \in \mathbb{R}^{BC\times T}$, and then passes it through a fully connected network to obtain the output U_t^l:

$$U_t^l = W_{t2}^l \sigma\left(\left(W_{t1}^l X_t^T\right)\right)^T \tag{10}$$

where $W_{t1}^l \in \mathbb{R}^{D_h\times T}$ and $W_{t2}^l \in \mathbb{R}^{T\times D_h}$ are the weight matrices of the fully connected layers, D_h is the hidden layer dimension, and σ is the activation function, with GELU used in this work.

The operation of the Feature Mixing sublayer is like that of the Time Mixing sublayer, except that the input features are reshaped into the form $X_c \in \mathbb{R}^{BT\times C}$, and then passed through two fully connected layers to obtain the output U_c^l:

$$U_c^l = W_{c2}^l \sigma(W_{c1}^l X_c) \tag{11}$$

Finally, through residual connections and Layer Normalization, the output of the l-th Mixer Layer X^l, is obtained:

$$X^l = \text{LayerNorm}\left(X^{(l-1)} + \text{Reshape}\left(U_t^l + U_c^l\right)\right) \tag{12}$$

where the Reshape$(\cdot)$ operation reshapes the tensor into the form $\mathbb{R}^{B\times T\times C}$.

In this work N stacked Mixer Layers are placed at the front of the Anomaly Transformer encoder, forming the preprocessing module for time-series data, referred to as MixerPretrain():

$$\text{MixerPretrain}(X) = X^l\left(X^{(l-1)}\left(\cdots X^1\cdots\right)\right) \tag{13}$$

where $X \in \mathbb{R}^{B\times T\times C}$ is the original multivariate time-series data, with C representing the original feature dimension, and $l \in \{1, 2, \ldots, N\}$. The output of MixerPretrain(X) contains time-series features with more global contextual information, which replace the original multivariate time-series data as input to the Anomaly Transformer encoder.

4.3. Adaptive Thresholds

Anomalies are inherently relative and should be dynamically determined based on context. For instance, in the SMAP dataset, higher soil moisture values may fall within the normal range during the rainy season but could be considered anomalous during the dry season. Similarly, in the MSL dataset, significant temperature differences exist between day and night, requiring separate anomaly detection for each. Fixed thresholds fail to capture these context-dependent anomaly patterns, making them inadequate for adapting to the complex characteristics of time-series data.

At each time t, the anomaly score sequence within a window of length w before and after time t, i.e., $\{S_{t-w}, \ldots, S_t, \ldots, S_{t+w}\}$, is observed. The calculation of the adaptive threshold is based on two fundamental assumptions:

1. Local Stationarity Assumption: Over short time windows, the statistical properties (such as mean and variance) of the time series remain relatively stable. This allows for the use of local statistics within the window to set the anomaly threshold. This assumption is consistent with the local variation characteristics observed in datasets like SMAP, MSL, and PSM, where indicators such as soil moisture and temperature tend to remain stable over short periods.
2. Anomaly Sparsity Assumption: The majority of time points are normal, with anomalies occurring infrequently. This suggests that quantiles can be used to define an upper bound for local anomalies. This assumption reflects the low-frequency occurrence of anomalies, such as drought events in the SMAP dataset and extreme weather events in the MSL dataset.

Based on the above assumptions, we adopt the Exponential Weighted Moving Average (EWMA) method to smooth the input values in order, to better capture the long-term trends in the data. Specifically, for a given anomaly score

S_t at the current time step, the corresponding EWMA value EWMA_t can be updated using the following recursive relation:

$$\mathrm{EWMA}_t = \alpha\,\mathrm{EWMA}_{(t-1)} + (1-\alpha)S_t \tag{14}$$

where $\mathrm{EWMA}_{(t-1)}$ is the Exponential Weighted Moving Average from the previous time step, and $\alpha \in (0,1)$ is the smoothing factor. A larger value of α means that the model places more emphasis on the most recent data points, while a smaller α value increases the dependence on historical data, resulting in a smoother model.

Additionally, an adaptive threshold update mechanism is designed to establish a sensitive response boundary based on the difference between the current input value and the moving average. The update formula for the threshold is expressed as follows:

$$\delta_t = \alpha\delta_{(t-1)} + (1-\alpha)\left(\mathrm{EWMA}_t + 2\sqrt{(S_t - \mathrm{EWMA}_t)^2}\right) \tag{15}$$

where δ_t is the threshold at the current time step. The term $\sqrt{(S_t - \mathrm{EWMA}_t)^2}$ calculates the absolute deviation between the current anomaly score S_t and the moving average value EWMA_t, representing the distance between the current observation and the smoothed trend.

In the threshold update process, not only is the threshold at the previous time step, δ_{t-1}, considered, but the current deviation, i.e., the absolute value of $(S_t - \mathrm{EWMA}_t)$, is also incorporated. This approach enables the threshold to adapt to fluctuations in the input data, thereby better reflecting changes in the current data.

4.4. Complexity Analysis and Optimization

Compared to the original Anomaly Transformer, our method effectively reduces the computational complexity of local dependency modeling by incorporating the Mixer architecture. Specifically, the Mixer component uses a local self-attention mechanism with a time complexity of $O(Nd)$, which significantly lowers the computational cost for processing long time series. In contrast, the Anomaly Transformer relies on a global self-attention mechanism with a time complexity of $O(L_{\text{Transformer}}N^2d)$, primarily due to the dependency calculations across the entire sequence. The overall time complexity of the combined model is $O(L_{\text{Mixer}}Nd) + O(L_{\text{Transformer}}N^2d)$, where the Mixer reduces the burden of local computations, but the global dependency modeling in the Anomaly Transformer still dominates the computational cost.

To reduce computational overhead, this paper proposes a soft sparsification method based on threshold filtering. By introducing a learnable threshold parameter, the model adaptively truncates attention scores according to their actual distribution. Although the resulting sparse pattern may lack strict regularity, it more accurately captures the intrinsic characteristics of the data. Furthermore, the sparsification degree is controlled by a sparsity coefficient, offering greater flexibility in balancing performance and efficiency.

The specific implementation is as follows: a learnable parameter τ is introduced prior to the *Softmax* operation to serve as a threshold for filtering attention scores.

$$\tilde{A} = \text{ReLU}\left(\frac{QK^\top}{\sqrt{D}} - \tau\right) \tag{16}$$

where $\tilde{A}$ is the sparse attention score matrix, and ReLU() denotes the rectified linear unit activation function. Attention scores below the threshold τ are truncated to zero, effectively filtering out weaker scores and enabling sparse control over the attention mechanism.

$$A = \beta\text{Softmax}(\tilde{A}) + (1-\beta)\text{Softmax}\left(\frac{QK^\top}{\sqrt{D}}\right) \tag{17}$$

where β is a controllable sparsity parameter that allows flexible adjustment of the sparsification intensity.

5. Algorithm and Execution Steps

In this section, the step-by-step process for implementing time series anomaly detection is outlined, utilizing a combination of Mixer-based preprocessing, association computation, and adaptive threshold updating. The proposed methodology is designed to capture subtle and dynamic anomalies in time series data, particularly in the presence of complex, interdependent features. The primary goal is to identify anomalous time points while dynamically adjusting the detection threshold based on the evolving characteristics of the data.

To achieve this, three main algorithmic execution steps are introduced. First, the original time series data is processed through an N-layer Mixer architecture to extract time- and feature-related patterns. This preprocessing step is crucial for transforming the raw data into a form that can be analyzed for anomalous behavior. Second, the preprocessed data is used to calculate two key association metrics—prior-association (PA) and series-association (SA)—which are then combined to compute the anomaly score. Finally, the adaptive threshold update step fine-tunes the anomaly score using an exponentially weighted moving average (EWMA) technique, allowing the model to adapt to changes in the data over time.

Time series anomaly detection is implemented through three algorithmic execution steps, as shown in Figure 3.

Algorithm 1 takes the input time series data $X \in \mathbb{R}^{B\times T\times C}$ and processes it through an N-layer Mixer structure. The time series data is reshaped at each layer, where it undergoes two main operations: *Time Mixing* and *Feature Mixing*. In the Time Mixing step, the model learns temporal dependencies by applying transformations along the time axis. Similarly, Feature Mixing operates on the feature dimension to capture dependencies among different features. Both operations involve matrix multiplication followed by nonlinear activation functions, such as ReLU or Sigmoid, introducing non-linearity to the model.

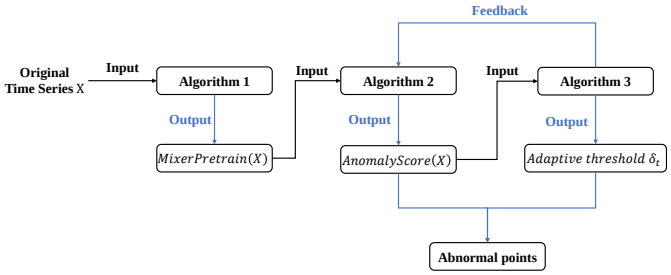

Figure 3: Algorithm Execution Steps.

Algorithm 1 Mixer Preprocessing Time Series

Input: Time series $X \in \mathbb{R}^{B \times T \times C}$, N=4
Output: MixerPretrain(X) # Time series after N layers of Mixer Layer processing
for $l = 1$ **to** N **do**
  Time Mixing:
  $X_t \leftarrow$ Time Mixing Sublayer-Reshape($X^{(l-1)}$)
  $U_t^l \leftarrow W_{t2}^l \cdot \sigma(W_{t1}^l X_t^T)^T$
  Feature Mixing:
  $X_c \leftarrow$ Feature Mixing Sublayer-Reshape($X^{(l-1)}$)
  $U_c^l \leftarrow W_{c2}^l \cdot \sigma(W_{c1}^l X_c)$ # $W_{c1}^l, W_{c2}^l \in \mathbb{R}^{BT \times C}$
  $X^l \leftarrow \text{LayerNorm}(X^{(l-1)} + \text{Reshape}(U_t^l + U_c^l))$
end for
Return: MixerPretrain(X) $\leftarrow X^l(X^{(l-1)}\,(\cdots X^1\,\cdots))$

After each operation, the results are combined using a residual connection to retain the information from previous layers, mitigating the vanishing gradient problem. Layer normalization is then applied to stabilize the training process and avoid overfitting. The final output, denoted as X^l, is a transformed version of the original time series data, which is then passed to the next step in the anomaly detection pipeline.

Algorithm 2 Anomaly Score Calculation

Input: MixerPretrain(X), where $X \in \mathbb{R}^{B \times T \times C}$
Output: AnomalyScore(X)
Compute PA(MixerPretrain(X))
Compute SA(MixerPretrain(X))
Compute AssDis(PA, SA; MixerPretrain(X))
Return: AnomalyScore(X)

Algorithm 2 calculates the anomaly score of the input time series data. It begins by processing the preprocessed time series data MixerPretrain(X)

to compute two key association measures: prior-association (PA) and series-association (SA). PA captures the temporal dependencies between adjacent time points, quantifying how much the current data depends on past data. On the other hand, SA measures the relationship between the current time point and other points across the entire time series, capturing long-term dependencies. The Association Discrepancy (AssDis) is then computed as the difference between PA and SA, serving as a key indicator of deviation from normal patterns. A high AssDis value suggests that the point is an anomaly. The final output is the Anomaly Score, which indicates the likelihood of each time point being an outlier, based on the computed associations and their discrepancies.

The purpose of Algorithm 3 is to dynamically adjust the anomaly detection threshold based on the calculated anomaly scores over time. First, the anomaly scores are sorted in ascending order to ensure they are processed in temporal order. The Exponential Weighted Moving Average (EWMA) is then computed for each score, with a smoothing factor α that gives more weight to recent values, allowing the threshold to adapt to the evolving data behavior. The updated threshold δ_t is computed by combining the previous threshold δ_{t-1} and the current EWMA value, along with an additional term accounting for the deviation between the anomaly score and the EWMA. This adaptive threshold mechanism enables the model to react to changing patterns in the data, ensuring that the anomaly detection remains sensitive and accurate over time. The final output is the set of detected anomaly points, identified as those whose anomaly scores exceed the updated threshold.

Algorithm 3 Adaptive Threshold Update

Input: AnomalyScore(X), w, α
Output: Updated AnomalyScore(X)
$\{S_1, \cdots, S_t, \cdots, S_T\} \leftarrow \text{Sort}(\text{AnomalyScore}(X))$
while t is valid **do**
  $S^w \leftarrow \{S_{t-w}, \cdots, S_t, \cdots, S_{t+w}\}$
  $\text{EWMA}_t \leftarrow \alpha \cdot \text{EWMA}_{t-1} + (1-\alpha) \cdot S_t$
  $\delta_t \leftarrow \alpha \cdot \delta_{t-1} + (1-\alpha) \cdot \left(\text{EWMA}_t + 2\sqrt{(S_t - \text{EWMA}_t)^2}\right)$
end while

6. Experiments

This section evaluates the performance of the proposed anomaly detection model across various datasets. The results demonstrate that the model outperforms existing methods when handling complex time series data, particularly in feature extraction and anomaly detection accuracy.

6.1. Experimental Setup

The experiments were conducted on an Ubuntu 20.04 operating system. The hardware environment utilized a Tesla V100-SXM2-16GB GPU (1 unit) and an

Intel Xeon Platinum 8163 @ 2.50GHz CPU (8 cores), with deep learning acceleration provided by CUDA 12.2 and cuDNN 8.7.0. The software environment consisted of Python 3.8.13 and PyTorch 2.2.0.

6.2. Datasets

Three public datasets were used for these experiments. The SMAP (Soil Moisture Active Passive) dataset, sourced from NASA's soil moisture monitoring mission, contains time-series data of soil moisture recorded by sensors. The anomalies in this dataset correspond to extreme weather events, such as droughts and floods, and are commonly used in environmental monitoring and anomaly detection research. The MSL (Mars Science Laboratory) dataset, provided by NASA's Mars Science Laboratory mission, records sensor data generated during the operation of the Mars rover. The anomalies in this dataset are related to extreme environmental conditions on the Martian surface, such as dust storms and extreme temperatures, making it suitable for equipment fault detection and health monitoring research. The third dataset is the PSM (Pooled Server Metrics), and contains time-series data of performance metrics collected from servers during their operation. The anomalies in this dataset span a wide range of real-world system failures and performance issues, including hardware errors, software bugs, and resource leaks, and is widely used in anomaly detection and server performance analysis.

The anomaly ratios in the three datasets range from 10.53% to 27.76%, providing a basis for evaluating the algorithm's performance under varying anomaly frequencies. Additionally, each dataset presents unique challenges, such as weak correlations in the SMAP, high coupling in the MSL, and a high anomaly rate in the PSM. These characteristics offer diverse testing scenarios for a comprehensive evaluation of the algorithm's performance.

In the experiments, each dataset was randomly split into training, validation, and test sets in the 3:1:1 ratio. The training set was used for model learning, the validation set for hyperparameter optimization and early stopping monitoring, and the test set for final performance evaluation.

6.3. Results

To systematically and comprehensively evaluate the proposed anomaly detection method, it is compared with 13 anomaly detection techniques. These methods represent a range of mainstream paradigms in the field, including support vector machine-based approaches, tree-based ensemble learning methods, density estimation-based methods, principal component analysis (PCA)-based subspace methods, autoregressive and long short-term memory (LSTM)-based predictive methods, as well as generative approaches based on variational autoencoders (VAE) and generative adversarial networks (GAN), among others.

As shown in the experimental results in Table 2 , the proposed anomaly detection model outperforms the comparison methods across the SMAP, MSL, and PSM datasets. On the SMAP dataset, the model achieves an F1 score of 97.49%, while on the MSL and PSM datasets, the F1 scores are 95.18%

Table 2: Comparative Experimental Results

Method	SMAP			MSL			PSM		
	P	R	F1	P	R	F1	P	R	F1
Deep-SVDD[31]	89.93	56.02	69.04	91.92	76.63	83.58	95.41	86.49	90.73
DAGMM [32]	86.45	56.73	68.51	89.60	63.93	74.62	93.49	70.03	80.08
MMPCACD [33]	88.61	75.84	81.73	81.42	61.31	69.95	76.26	78.35	77.29
LSTM [34]	89.41	78.13	83.39	85.45	82.50	83.95	76.93	89.64	82.80
CL-MPPCA [17]	86.13	63.16	72.88	73.71	88.54	80.44	56.02	99.93	71.80
ITAD [35]	82.42	66.89	73.85	69.44	84.09	76.07	72.80	64.02	68.13
LSTM-VAE [36]	92.20	67.75	78.10	85.49	79.94	82.62	73.62	89.92	80.96
BeatGAN [37]	92.38	55.85	69.61	89.75	85.42	87.53	90.30	93.84	92.04
OmniAnomaly[38]	92.49	81.99	86.92	89.02	86.37	87.67	88.39	74.46	80.83
InterFusion [39]	89.77	88.52	89.14	81.28	92.70	86.62	83.61	83.45	83.52
THOC [40]	92.06	89.34	90.68	88.45	90.97	89.69	88.14	90.99	89.54
Anomaly-Transformer [20]	94.06	99.27	96.59	92.05	94.50	93.26	97.37	98.27	97.82
MEMTO [41]	93.76	**99.63**	96.61	92.07	96.76	94.36	97.46	**99.23**	**98.34**
Mixer-Transformer	**97.53**	98.47	**97.49**	**93.53**	**96.89**	**95.18**	**97.55**	98.85	98.20

and 98.20%, respectively. These gains are primarily attributed to two innovations within the Transformer framework. First, the introduction of the Mixer structure at the frontend of the Transformer encoder significantly enhances the model's ability to capture global dependencies and behavioral patterns in time-series data. The Mixer captures multi-scale dynamic features, such as trends, cycles, and fluctuations, enabling more comprehensive feature extraction from complex time-series data. Second, the adaptive threshold, derived from statistical features, automatically adjusts the anomaly detection criteria, improving the model's accuracy and robustness in handling non-stationary data variations.

Compared to traditional machine learning methods, such as OC-SVM, Isolation Forest, and LOF, the proposed method significantly outperforms in terms of precision, recall, and in its F1 score on all three datasets, with improvements ranging from 20% to 40%. This demonstrates that deep learning-based anomaly detection models are better at capturing complex and abstract feature representations in time-series data, offering stronger anomaly pattern characterization and generalization capabilities. Traditional methods struggle to effectively model long-term dependencies in time-series data and fail to fully exploit interactions between multiple variables, limiting their detection performance.

Furthermore, compared to classical time-series anomaly detection algorithms such as MMPCACD, VAR, and LSTM, the proposed method also achieves significant performance improvements. For example, while LSTM can capture long-term dependencies in time-series data, its recurrent network structure has limitations when modeling the complex interactions between different time steps and variables. In contrast, the proposed model's Mixer-Transformer architecture enhances the learning capability of spatiotemporal features. This highlights the advantages of alternating modeling of the channel and time dimensions, as well as the self-attention mechanism, in capturing global dependencies in time-series data.

Additionally, the proposed method was compared with several recently introduced deep anomaly detection models, such as CL-MPPCA, ITAD, LSTM-

VAE, and BeatGAN. The results show that the proposed method outperforms these models across all three datasets. For example, despite using the advanced Generative Adversarial Network (GAN) paradigm, BeatGAN's performance is still inferior to that of the proposed model. This is primarily due to the inherent challenges in training and tuning GAN models, which can lead to less stable anomaly score estimations. In contrast, the proposed model leverages the self-attention mechanism and incorporates an adaptive threshold strategy, enhancing its anomaly detection capability and leading to overall performance improvements.

Table 3: F1 Score of Mixer Layers of Three Datasets

Mixer Layer	SMAP	MSL	PSM
0	95.46	92.45	97.36
1	95.82	93.65	97.85
2	96.04	93.15	98.06
3	96.62	94.69	97.92
4	**97.49**	**95.18**	**98.20**
5	96.36	94.83	0
6	96.21	94.75	0
7	96.26	94.73	0
8	96.1	95.2	0
9	96.01	95.22	0

As part of the analysis, the effect of the number of Mixer layers on the model's performance is examined. As shown in Table 3, optimal performance is achieved when the number of Mixer layers is set to 4 for all three datasets. However, when the number of layers exceeds 4, the F1 scores for the PSM datasets drop to zero.

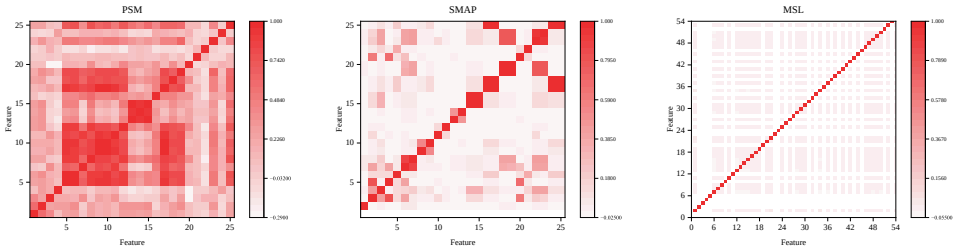

Figure 4: Heatmaps of Feature Correlations of the Three Datasets.

As shown in Figure 4 , the features of the SMAP dataset exhibit weak correlations, with limited negative correlations and a relatively simple overall structure. As a result, when the number of Mixer layers exceeds 4, the model can still effectively extract relevant features and maintain stable performance. In contrast, the MSL dataset features strong correlations and a more complex clustering structure, allowing the model to capture significant global feature in-

teractions as the number of layers increases. However, an excessive number of layers introduces redundant information, leading to a slight decrease in performance, though not a drastic one. In comparison, the PSM dataset presents the highest feature complexity, containing diverse correlation patterns, including numerous negative correlations (e.g., resource competition). These correlations make the model prone to introducing noise as the number of layers increases, resulting in overfitting of conflicting relationships between complex features. Additionally, as the number of layers increases, the gradient propagation path lengthens, exacerbating training instability and ultimately causing the model's performance to deteriorate to 0.

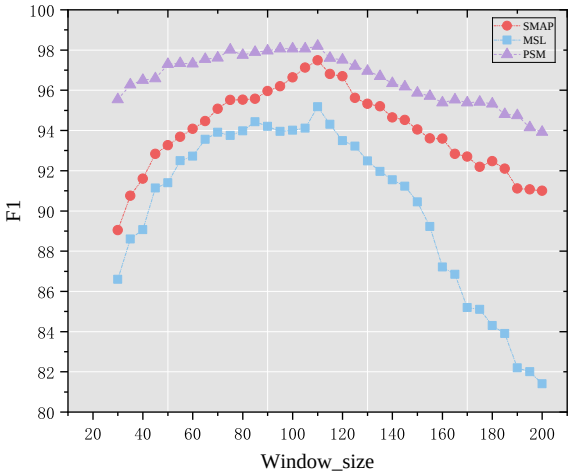

Figure 5: Impact of the Window Size on F1 Score for the Three Datasets.

Additional experiments were conducted to evaluate the effect of window size, with the range set from 30 to 200. As shown in Figure 5, a window size of 110 yielded the best performance across the three datasets. For the MSL dataset, due to the presence of multiple highly correlated feature clusters and specific temporal relationships, these patterns are only valid within a certain window size. When the window size is too large (e.g., 150), the structure of these feature clusters can be disrupted, leading to a significant decline in performance.

6.4. Ablation

To comprehensively evaluate the impact and contribution of the two key innovations proposed in this paper—the Mixer structure and the adaptive threshold—ablation experiments were conducted. The results of these experiments are presented in Table 4 below:

The results of the ablation experiments highlight the significant contributions of both the Mixer structure and the adaptive threshold to the model's performance. As shown in Table 4 and Figure 6 :

Impact of the Mixer structure: Incorporating the Mixer structure leads to substantial improvements in model performance, particularly in feature representation and in modeling global dependencies. On the SMAP dataset, the F1 score increased from 96.59% to 97.01%, while on the MSL and PSM datasets, the F1 scores improved from 93.26% and 97.82% to 94.40% and 97.92%, respectively. These improvements suggest that the Mixer structure effectively captures the dynamic and complex characteristics of time series by alternating between modeling interactions along the channel and time dimensions.

Table 4: Ablation Experiment Results

Method	SMAP			MSL			PSM		
	P	R	F1	P	R	F1	P	R	F1
Baseline (B)	94.06	99.27	96.59	92.05	94.50	93.26	97.37	98.27	97.82
Baseline + Mixer (B+M)	95.96	98.08	**97.01**	93.90	94.90	**94.40**	97.20	98.65	**97.92**
Baseline + Adaptive threshold (B+A)	94.55	99.20	**96.82**	93.12	94.89	**94.00**	97.38	98.66	**98.02**
Baseline + Mixer + Adaptive threshold (B+M+A)	96.53	98.47	**97.49**	93.53	96.89	**95.18**	97.55	98.85	**98.20**

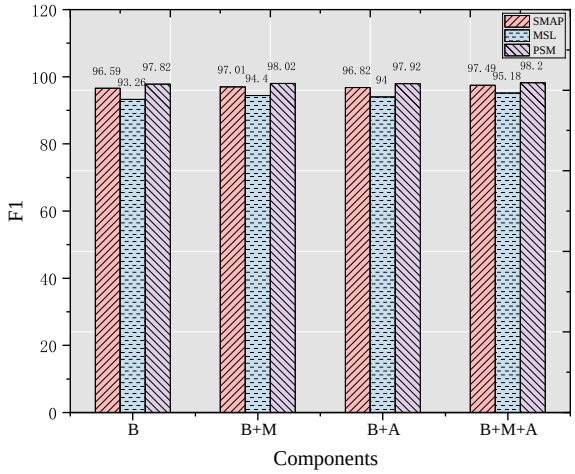

Figure 6: Impact of Mixer Structure and the Adaptive Threshold on Model Performance.

Impact of the Adaptive threshold: The introduction of the adaptive threshold further enhances the model's ability to handle non-stationary time series data. On the SMAP, MSL, and PSM datasets, the F1 scores improved from 96.59%, 93.26%, and 97.82% to 96.82%, 94.00%, and 98.02%, respectively. The adaptive threshold dynamically adjusts the anomaly detection standard based on the local distribution of anomaly scores, enabling the model to flexibly adapt to variations in different time intervals. This results in greater robustness and accuracy in detection.

Combined Impact: When the Mixer structure and the adaptive threshold are combined, the model achieves optimal performance. The F1 scores on the SMAP, MSL, and PSM datasets reach 97.49%, 95.18%, and 98.20%, re-

spectively, surpassing the individual improvements from each innovation. This demonstrates that the two components complement each other, with the Mixer enhancing feature extraction and interaction, and the adaptive threshold adding flexibility in dynamic adjustment. Together, they contribute to the overall enhancement of the model's performance and stability.

Compared to the baseline, the Mixer structure demonstrates a more significant improvement in detection performance than the adaptive threshold, owing to its strong feature fusion capability.

Table 5: Train and Test Time Comparison Across the Three Datasets

Dataset	Method	Train(s)	Test(s)
SMAP	Anomaly-Transformer[20]	699.03	29.51
	Mixer-Transformer	718.66	28.68
MSL	Anomaly-Transformer[20]	483.20	28.44
	Mixer-Transformer	491.98	27.32
PSM	Anomaly-Transformer[20]	476.87	27.96
	Mixer-Transformer	482.11	27.28

Table 5 presents a comparative analysis of the training and testing times of Anomaly-Transformer [20] and the proposed Mixer-Transformer across three datasets: SMAP, MSL, and PSM. The results show that Anomaly-Transformer achieves lower training times—699.03s, 483.20s, and 476.87s on SMAP, MSL, and PSM, respectively—compared to 718.66s, 491.98s, and 482.11s for Mixer-Transformer, reflecting an increase of 1.1% to 2.8% in training duration. This increase is attributed to the additional computational complexity introduced by the mixer's structure. However, by incorporating a coefficient attention mechanism, Mixer-Transformer demonstrates improved testing efficiency, with inference times of 28.68s, 27.32s, and 27.88s on SMAP, MSL, and PSM, respectively, outperforming Anomaly-Transformer's 29.51s, 28.44s, and 27.96s by 0.3% to 3.9%. The most significant improvement in testing time is observed on the MSL dataset, while the largest training time disparity occurs on SMAP.

These results highlight the effectiveness of the coefficient attention mechanism in enhancing inference speed, suggesting that Mixer-Transformer is a promising approach for real-time anomaly detection despite its slightly increased training overhead. Future work will focus on further optimizing the training process to reduce computational costs while maintaining the benefits in inference efficiency.

7. Conclusion

This study addresses the challenges in time series anomaly detection and proposes an innovative method that integrates multivariate feature fusion with adaptive threshold adjustment. While time series anomaly detection plays a critical role across various fields, traditional methods are often constrained by the nonlinear and non-stationary nature of the data, as well as the rigidity of

fixed threshold settings. The proposed Mixer-Transformer method introduces the Mixer architecture to model global dependencies across both channel and time dimensions, thereby enhancing the system's ability to capture complex dynamic features. Additionally, an adaptive threshold update mechanism, based on the sliding window approach and the exponentially weighted moving average method, is employed to better accommodate dynamic data variations, improving anomaly detection accuracy. Future work in this area will focus on anomaly causal analysis to uncover the underlying causes of time series anomalies. By exploring the potential causal relationships behind abnormal events, this study aims to more accurately pinpoint anomaly sources and enhance the interpretability and decision-making capabilities of anomaly detection systems.

Acknowledgments

This research was funded by the Key Research and Development Program of Zhejiang Province No.2023C01141, the Science and Technology Innovation Community Project of Yangtze River Delta No.23002410100, the Key Research Project of Zhejiang Province No.2024C01212, and the Department of Education Research Project of Zhejiang Province (Y202044560), the Ministry of Education Industry-University Collaboration Coordinated Education Project of China (221006521142515).

This work was supported by the Open Research Fund of The State Key Laboratory of Blockchain and Data Security, Zhejiang University.

References

[1] D. Manivannan, Recent endeavors in machine learning-powered intrusion detection systems for the internet of things, Journal of Network and Computer Applications (2024) 103925.

[2] N. Bugshan, I. Khalil, M. S. Rahman, M. Atiquzzaman, X. Yi, S. Badsha, Toward trustworthy and privacy-preserving federated deep learning service framework for industrial internet of things, IEEE Transactions on Industrial Informatics 19 (2) (2022) 1535–1547.

[3] A. Mahboubi, K. Luong, H. Aboutorab, H. T. Bui, G. Jarrad, M. Bahutair, S. Camtepe, G. Pogrebna, E. Ahmed, B. Barry, et al., Evolving techniques in cyber threat hunting: A systematic review, Journal of Network and Computer Applications (2024) 104004.

[4] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, M. Salehi, Deep learning for time series anomaly detection: A survey, ACM Computing Surveys 57 (1) (2024) 1–42.

[5] C. Wang, K. Wu, T. Zhou, G. Yu, Z. Cai, Tsagen: synthetic time series generation for kpi anomaly detection, IEEE Transactions on Network and Service Management 19 (1) (2021) 130–145.

[6] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, K. Veeramachaneni, Tadgan: Time series anomaly detection using generative adversarial networks, in: 2020 ieee international conference on big data (big data), IEEE, 2020, pp. 33–43.

[7] N. Bugshan, I. Khalil, A. P. Kalapaaking, M. Atiquzzaman, Intrusion detection-based ensemble learning and microservices for zero touch networks, IEEE Communications Magazine 61 (6) (2023) 86–92.

[8] P. Bidyuk, A. Gozhyj, I. Kalinina, V. Vysotska, Methods for forecasting nonlinear non-stationary processes in machine learning, in: International Conference on Data Stream Mining and Processing, Springer, 2020, pp. 470–485.

[9] Y. Luo, Y. Wang, A statistical time-frequency model for non-stationary time series analysis, IEEE Transactions on Signal Processing 68 (2020) 4757–4772.

[10] K. Zhou, W. Wang, T. Hu, K. Deng, Time series forecasting and classification models based on recurrent with attention mechanism and generative adversarial networks, Sensors 20 (24) (2020) 7211.

[11] P. Arumugam, R. Saranya, Outlier detection and missing value in seasonal arima model using rainfall data, Materials Today: Proceedings 5 (1) (2018) 1791–1799.

[12] X. Fang, W. Zhang, J. Lin, Y. Liu, Research on sdn fingerprint attack defense mechanism based on dynamic disturbance and information entropy detection, Security and Communication Networks 2022 (1) (2022) 1957497.

[13] J. da Silva Arantes, M. da Silva Arantes, H. B. Fröhlich, L. Siret, R. Bonnard, A novel unsupervised method for anomaly detection in time series based on statistical features for industrial predictive maintenance, International journal of data science and analytics 12 (4) (2021) 383–404.

[14] 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.

[15] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[16] Y. Wang, X. Du, Z. Lu, Q. Duan, J. Wu, Improved lstm-based time-series anomaly detection in rail transit operation environments, IEEE Transactions on Industrial Informatics 18 (12) (2022) 9027–9036.

[17] S. Tariq, S. Lee, Y. Shin, M. S. Lee, O. Jung, D. Chung, S. S. Woo, Detecting anomalies in space using multivariate convolutional lstm with mixtures of probabilistic pca, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2123–2133.

[18] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, C. Wu, Arrhythmia classification of lstm autoencoder based on time series anomaly detection, Biomedical Signal Processing and Control 71 (2022) 103228.

[19] C.-L. Liu, W.-H. Hsaio, Y.-C. Tu, Time series classification with multivariate convolutional neural network, IEEE Transactions on industrial electronics 66 (6) (2018) 4788–4797.

[20] J. Xu, Anomaly transformer: Time series anomaly detection with association discrepancy, arXiv preprint arXiv:2110.02642 (2021).

[21] Q. Ni, X. Cao, Mbgan: An improved generative adversarial network with multi-head self-attention and bidirectional rnn for time series imputation, Engineering Applications of Artificial Intelligence 115 (2022) 105232.

[22] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, Y. Liu, Tempo: Prompt-based generative pre-trained transformer for time series forecasting, arXiv preprint arXiv:2310.04948 (2023).

[23] K. Doshi, S. Abudalou, Y. Yilmaz, Tisat: Time series anomaly transformer, arXiv preprint arXiv:2203.05167 (2022).

[24] Y. Chen, F. Xing, S. Ma, Z. A. Bhuiyan, S. Lei, X. Jing, Trackspear: Attention-guided adversarial patches for probing security in visual tracking, TechRxiv preprint TechRxiv:173612760.01399750 (2025).

[25] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, T. Pfister, Tsmixer: An all-mlp architecture for time series forecasting, arXiv preprint arXiv:2303.06053 (2023).

[26] C.-Y. Hsu, W.-C. Liu, Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing, Journal of Intelligent Manufacturing 32 (3) (2021) 823–836.

[27] X. Wang, D. Pi, X. Zhang, H. Liu, C. Guo, Variational transformer-based anomaly detection approach for multivariate time series, Measurement 191 (2022) 110791.

[28] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, arXiv preprint arXiv:2201.07284 (2022).

[29] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 4027–4035.

[30] C. Lin, B. Du, L. Sun, L. Li, Hierarchical context representation and self-adaptive thresholding for multivariate anomaly detection, IEEE Transactions on Knowledge and Data Engineering (2024).

[31] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International conference on machine learning, PMLR, 2018, pp. 4393–4402.

[32] 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.

[33] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, N. Takata, A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction, IEEE Transactions on Aerospace and Electronic Systems 53 (3) (2017) 1384–1401.

[34] 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, 2018, pp. 387–395.

[35] Y. Shin, S. Lee, S. Tariq, M. S. Lee, O. Jung, D. Chung, S. S. Woo, Itad: integrative tensor-based anomaly detection system for reducing false positives of satellite systems, in: Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 2733–2740.

[36] D. Park, Y. Hoshi, C. C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robotics and Automation Letters 3 (3) (2018) 1544–1551.

[37] B. Zhou, S. Liu, B. Hooi, X. Cheng, J. Ye, Beatgan: Anomalous rhythm detection using adversarially generated time series., in: IJCAI, Vol. 2019, 2019, pp. 4433–4439.

[38] 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, 2019, pp. 2828–2837.

[39] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, D. Pei, Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding, in: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, 2021, pp. 3220–3230.

[40] L. Shen, Z. Li, J. Kwok, Timeseries anomaly detection using temporal hierarchical one-class network, Advances in Neural Information Processing Systems 33 (2020) 13016–13026.

[41] J. Song, K. Kim, J. Oh, S. Cho, Memto: Memory-guided transformer for multivariate time series anomaly detection, Advances in Neural Information Processing Systems 36 (2023) 57947–57963.