Expert Systems With Applications 271 (2025) 126667

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Graph-enhanced anomaly detection framework in multivariate time series using Graph Attention and Enhanced Generative Adversarial Networks

Yue He [a], Xiaoliang Chen [a,b,e,*], Duoqian Miao [b], Hongyun Zhang [b], Xiaolin Qin [c], Shangyi Du [d], Peng Lu [e]

[a] School of Computer and Software Engineering, Xihua University, Chengdu 610039, PR China
[b] College of Electronic and Information Engineering, Tongji University, Shanghai 201804, PR China
[c] Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, PR China
[d] Department of Computer Science, Mcgill University, Montreal, QC, H3A0G4, Canada
[e] Department of Computer Science and Operations Research, University of Montreal, Montreal, QC, H3C3J7, Canada

ARTICLE INFO

Keywords:
Time series
Anomaly detection
Graph attention mechanism
Generative adversarial networks
Cloud computing networks

ABSTRACT

The complexity and scale of distributed systems in cloud computing present significant challenges for effective time series anomaly detection, which aims to identify unusual patterns in time series data that deviate from expected behavior. Traditional anomaly detection technologies in this domain suffer from high false positive rates. This challenge arises from the difficulty of balancing high recall rates with the reduction of false positives, which are both essential for ensuring operational integrity and user satisfaction in cloud environments. To address these challenges, this paper presents the Efficient Hybrid Graph Attention Mechanism and Enhanced Generative Adversarial Network (EH-GAM-EGAN), an innovative unsupervised model tailored for multivariate time series anomaly detection in cloud computing networks. First, EH-GAM-EGAN utilizes a graph attention mechanism combined with Long Short-Term Memory networks to effectively capture and analyze complex node relationships, thereby improving the understanding of data interdependencies. Second, it integrates an enhanced generative adversarial network, which precisely computes reconstruction and discrimination errors. This approach facilitates a thorough analysis of anomalies by examining reconstruction, discrimination, and prediction errors, resulting in significantly improved detection accuracy and model reliability. Extensive experiments on four publicly available cloud computing datasets empirically validated the effectiveness of EH-GAM-EGAN. The results show that EH-GAM-EGAN achieved average improvements of 17.93%, 17.88%, and 21.46% in precision, recall, and F1 scores, respectively.

1. Introduction

The digital transformation era, defined by the exponential growth of data and rapid advancements in artificial intelligence (AI), has underscored the critical role of cloud computing networks in both academic research and industrial applications. These networks, with their intricate and dynamic infrastructures, serve as the foundation for numerous essential services. However, their complexity makes them vulnerable to physical resource failures and anomalies, leading to performance degradation, operational inefficiencies, and diminished end-user experiences. Addressing these challenges has elevated the development of advanced anomaly detection technologies specifically designed for cloud computing networks into a pivotal area of research (Huang et al., 2022).

The use of monitoring tools allows for the collection of performance metrics, such as CPU and memory usage, in the form of multivariate time series data (Aslanpour, Gill, & Toosi, 2020). This data provides valuable insights into the health of the network, enabling the real-time detection of abnormal behaviors that may signal underlying issues. However, anomaly detection in this context is inherently challenging. The complex network of components and their interactions within cloud computing infrastructure complicates the identification of abnormal patterns. Additionally, the noise and uncertainty in time series data — caused by sensor inaccuracies, network delays, and dynamic resource allocations — further hinder the differentiation between normal and anomalous behaviors. This often leads to increased false positive rates, where normal behavior is mistakenly flagged as anomalous, reducing the reliability of detection technologies (Li & Jung, 2023).

* Corresponding author at: School of Computer and Software Engineering, Xihua University, Chengdu 610039, PR China.
E-mail addresses: hey@stu.xhu.edu.cn (Y. He), chenxl@mail.xhu.edu.cn, chexiaol@iro.umontreal.ca (X. Chen), dqmiao@tongji.edu.cn (D. Miao), zhanghongyun@tongji.edu.cn (H. Zhang), qinxl@casit.com.cn (X. Qin), shangyi.du@mail.mcgill.ca (S. Du), peng.lu@umontreal.ca (P. Lu).

https://doi.org/10.1016/j.eswa.2025.126667
Received 25 November 2024; Received in revised form 16 January 2025; Accepted 22 January 2025
Available online 4 February 2025

Traditional time series anomaly detection methods in cloud computing predominantly rely on statistical approaches. Rish et al. (2001) proposed a Naive Bayes method for detecting anomalies in univariate time series within cloud environments, while Palmieri, Fiore, and Castiglione (2014) applied independent component analysis for anomaly detection in univariate time series in cloud computing contexts.

Modern time series anomaly detection techniques in cloud computing predominantly rely on unsupervised deep learning approaches. Notably, LSTM-VAE (Park, Hoshi, & Kemp, 2018) replaces the fully connected layer in the Variational Autoencoder (VAE) with Long Short-Term Memory (LSTM), utilizing variational inference for multivariate sequence reconstruction. For univariate cloud time series anomaly detection, SR and SR-CNN (Ren et al., 2019) employ spectral residuals, a method originally developed for visual saliency detection. These methods are specifically designed for univariate Key Performance Indicator (KPI) data. OmniAnomaly (Su et al., 2019) adopts a stochastic recurrent neural network framework to catch intricate long-term temporal patterns, employing a variational autoencoder architecture to compute reconstruction probabilities.

Graph Neural Networks (GNN) have emerged as a potent solution to the limitations inherent in traditional deep learning models. For instance, Deng and Hooi (2021) propose a methodology that combines structure learning and GNN. The method accurately detects anomalies, captures sensor correlations effectively. Similarly, STGCN (Yu, Yin, & Zhu, 2018) consists of multiple spatio-temporal convolutional blocks, each structured in a "sandwich" configuration, with two gated sequential convolution layers surrounding a spatial graph convolution layer. This architecture allows STGCN to capture intricate spatio-temporal dependencies, effectively tackling time series prediction challenges in traffic forecasting.

Nevertheless, the existing research indicates that most methods developed for cloud computing scenarios are primarily focused on univariate time series data. Current graph models fail to effectively capture temporal dependencies. Moreover, present anomaly detection techniques leveraging deep learning primarily aim to enhance accuracy within specific contexts, often overlooking the complex and dynamic characteristics of cloud computing systems (Song et al., 2023). In light of these circumstances, the development of an unsupervised deep learning approach tailored for cloud computing environments is critically needed. This approach must be capable of not only accurately detecting anomalies in multivariate time series data but also robustly addressing the unique challenges posed by cloud computing networks to ensure reliable detection.

This study presents an innovative unsupervised model, termed the Efficient anomaly detection in multivariate time series using a Hybrid model with Graph Attention Mechanism and Enhanced Generative Adversarial Network (EH-GAM-EGAN). This model combines the capabilities of graph neural networks and enhanced generative adversarial networks (GAN) to create a comprehensive anomaly detection framework. GNN, with their proficiency in managing graph-structured data and elucidating relationships between nodes, are adept at uncovering the intricate interconnections within multivariate time series data (Ho, Karami, & Armanfard, 2023). Integrating a multi-head self-attention mechanism facilitates contextual information extraction from sequential data, while Long Short-Term Memory networks capture enduring temporal dependencies. Concurrently, the generative model within the GAN framework identifies anomalies by reconstructing data points or sequences from test data and comparing these reconstructions with actual values to compute reconstruction errors. The discriminative model serves to further enhance the efficacy of the model by distinguishing between instances derived from training data and those generated by the model, thereby generating discrimination errors.

The main contributions of this paper can be summarized as follows:

- **Graph Attention and GAN Integration for Anomaly Detection:** This study introduces the integration of a graph attention mechanism with enhanced GAN. By employing a graph attention mechanism, the model achieves a nuanced understanding of node relationships, enabling a comprehensive exploration of interdependencies among nodes. This is further augmented by the amalgamation of advanced deep learning methodologies, including multi-dimensional data feature extraction and the bolstering of anomaly detection robustness. The real-time adjustment of attention weights and adaptability to data variations significantly advance anomaly detection performance.
- **Error Analysis Framework:** A novel framework for integrating reconstruction, discrimination, and prediction errors has been developed, in which the prediction error is bifurcated into components derived from both the graph attention mechanism and the LSTM network. This approach has the dual objective of refining the precision of anomaly detection while concurrently minimizing false alarm rates.
- **Optimized Model Efficiency:** The EH-GAM-EGAN model distinguishes itself through its competitive advantage in reduced time complexity and accelerated training speed per iteration. This efficiency makes the model particularly well-suited for processing large datasets, addressing one of the critical challenges in anomaly detection. The optimization of model efficiency without compromising on detection accuracy represents a significant stride towards scalable and practical anomaly detection solutions.
- **Empirical Validation and Impact:** The effectiveness of EH-GAM-EGAN has been rigorously validated through experimental evaluations conducted on public datasets. These evaluations underscore the model's potential to significantly advance the state-of-the-art in anomaly detection, demonstrating its applicability and impact in real-world scenarios.

This paper is organized into several sections. Section 2 provides a summary of the relevant literature; Section 3 presents a detailed definition of the problem to be solved; Section 4 describes the approach taken to solve the problem; Section 5 outlines the experimental setup; and finally, in Section 6 summarizes the findings and presents concluding remarks.

2. Related work

The realm of identifying anomalies in multivariate time series data has been significantly expanded by a multitude of studies, each of which has contributed novel algorithms and methodologies. This section outlines the spectrum of approaches that are either directly relevant or have been instrumental in shaping the framework proposed in this study.

2.1. Multivariate time series anomaly detection

Multivariate time series anomaly detection (MTSAD) has garnered significant attention due to its crucial role in identifying atypical samples that deviate from expected patterns within datasets. This capability is essential across various domains, including cloud computing, finance, and healthcare, where early detection of anomalies can mitigate potential losses or adverse outcomes. Over the years, numerous methodologies have been proposed, each uniquely advancing the field. Traditional techniques have provided the foundational basis for anomaly detection, employing diverse approaches such as density-based methods (Breunig, Kriegel, Ng, & Sander, 2000), linear model-based approaches (Shyu, Chen, Sarinnapakorn, & Chang, 2003), distance-based methods (Angiulli & Pizzuti, 2002), classification models (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001), and ensemble detectors (Lazarevic & Kumar, 2005). These foundational methods have been instrumental in shaping the progression of this field, offering critical

concepts that underpin modern innovations.

In recent years, remarkable advances have been made in enhancing anomaly detection in high-dimensional datasets by applying deep learning techniques. One such advancement is CARLA (Darban, Webb, Pan, Aggarwal, & Salehi, 2025a) is a end-to-end self-supervised contrastive representation learning method for time-series anomaly detection. This research demonstrates the significant potential of contrastive representation learning in advancing the field of time-series anomaly detection. Darban, Webb, Pan, Aggarwal, and Salehi (2025b) have conducted a comprehensive categorization of deep models for time series anomaly detection based on their primary approaches and architectural designs. These approaches predominantly include prediction-based (Ergen & Kozat, 2019; Tealab, 2018) and reconstruction-based (Chen et al., 2021; Goodge, Hooi, Ng, & Ng, 2021) methodologies. The former is based on the premise of utilizing models to predict future values in a series, thereby identifying anomalies through deviations from these predictions. The latter, in contrast, focuses on reconstructing the time series data to pinpoint anomalies by evaluating the reconstruction error.

Prediction-based models are adept at forecasting future points or subsequences by examining current or adjacent data points. This approach predicates the anomaly detection mechanism on the disparity between predicted and actual values, with significant deviations flagging potential anomalies (Çavdar, Ebrahimpour, Kakız, & Günay, 2023). These models employ a sliding window technique to predict subsequent data points, facilitating the identification of anomalies by contrasting them against normative data patterns. This methodology's strength lies in its ability to model and predict normal behavior, rendering it particularly effective in scenarios where anomalies are sparse.

Among the various architectures, Recurrent Neural Networks (RNN) and their derivatives, such as LSTM and Gated Recurrent Units (GRU), have demonstrated superior performance in capturing temporal dependencies within time series data (Cho et al., 2014). Building on this foundation, Khanmohammadi and Azmi (2024) proposed the D-CNN-LSTM Autoencoder, a model designed for multi-sensor connectivity in connected and autonomous vehicles. This approach holds significant promise for enhancing smart transportation systems and addressing security and privacy challenges. Shanmuganathan and Suresh (2024) introduced the Markov-enhanced I-LSTM approach, which delivers unparalleled accuracy in predicting short- and long-term sensor data. Tang, Xu, Yang, Tang, and Zhao (2023) introduced the GRN, a method that combines neural graph networks and GRU to enhance the safety and reliability of industrial control systems. This approach offers substantial value and potential in anomaly detection. Deng et al. (2024) introduced SCNN, an adaptive, interpretable, and scalable forecasting framework that models each component of spatial–temporal patterns independently. SCNN operates based on a predefined MTS generative process, which mathematically captures the latent structure of spatial–temporal patterns, providing enhanced traceability and predictability compared to the original MTS.

Reconstruction-based models are popular for detecting anomalies in time series data. These models generally employ GAN to recreate time series data, with anomalies being identified based on the reconstruction error. Specifically, GAN reconstruct test data points or sliding windows and compares them with the actual values. This comparison generates reconstruction errors. When the probability of reconstructing anomalous points falls below a pre-defined threshold, anomalies are identified. GAN have been used to great effect in time series analysis, due to their ability to simplify training and sample generation without relying on Markov chains or unfolded approximate inference networks. This effectiveness has given rise to numerous GAN-based methods that have significantly advanced the field of MTSAD. Bashar and Nayak (2020) introduced TAnoGan, a Generative Adversarial Network-based approach for detecting anomalies in time series datasets with limited data points. Its remarkable performance highlights its efficiency under data scarcity and offers a novel solution for time series anomaly detection. Wen et al. (2022) presented a convolutional adversarial model integrating anomaly detection and explanation framework for identifying anomalies in multivariate time series data in cloud environments. Audibert, Michiardi, Guyard, Marti, and Zuluaga (2020) introduce USAD, an efficient and reliable method for UnSupervised Anomaly Detection in multivariate time series. Its unique architecture and utilization of adversarial training enable effective identification and isolation of anomalies, ensuring efficient training. Li et al. (2019) introduce MAD-GAN, an unsupervised multivariate anomaly detection approach harnessing GAN. LSTM-RNN within the GAN framework capture temporal correlations in time series distributions. MAD-GAN integrates a novel anomaly score called DR-score for discrimination and reconstruction-based anomaly detection. Chen et al. (2021) introduce a robust unsupervised anomaly detection framework that performs well across diverse datasets. The framework employs two discriminators to adversarially train an autoencoder, enabling it to learn the normal pattern of multivariate time series. Anomalies are detected based on the reconstruction error. Chen et al. (2024) introduce MACE, an efficient anomaly detection method in the frequency domain for time-series data, which adapts to multiple normal patterns. The method incorporates three novel features: a pattern extraction mechanism, a dual convolution mechanism, and the exploitation of sparsity and parallelism in the frequency domain to enhance model efficiency. MACE effectively handles diverse normal patterns using a unified model and achieves excellent performance with high efficiency. Xu, Wu, Wang, and Long (2022) argue that anomalous time points are unlikely to form strong associations with the entire time series due to their rarity, whereas normal time points exhibit the opposite behavior. To address this issue, AnomalyTrans employs a transformer equipped with an anomaly attention mechanism, which captures the differential associations of time points, thereby improving anomaly detection and achieving high accuracy. Yang, Zhang, Zhou, Wen, and Sun (2023) propose an anomaly detection model that normalizes multivariate time series and utilizes a dual attention mechanism to capture multiple perspectives. The model detects anomalies by measuring discrepancies in representations and calculates anomaly scores based on these differences, ensuring accurate detection. Zhou, Dai, Wang, and Qiu (2024) proposed GLAD, a Gumbel-Softmax-based graph structure learning strategy that effectively captures global topological associations to detect anomalies by leveraging differences between global and local associations. This approach demonstrates significant potential and innovation in anomaly detection.

Furthermore, the previously mentioned prediction-based and reconstruc
tion-based models exhibit limitations in capturing temporal dependencies or solely focusing on learning input feature representations without explicitly incorporating feature relationships. The EH-GAM-EGAN model, introduced in this paper, addresses the aforementioned issues by employing LSTM to effectively capture temporal dependencies and leveraging the Wasserstein distance with gradient penalty in the GAN framework to overcome convergence challenges. This novel approach enhances the model's ability to detect time series data anomalies. It guarantees a stable and efficient training process, representing a significant advancement in the field of MTSAD.

2.2. Graph neural networks

The exploration of MTSAD has recently been enriched by the advent of GNN, which have emerged as a potent solution to the limitations inherent in traditional deep learning models. Specifically, GNN excel in capturing the complex inter-feature relationships that conventional prediction-based and reconstruction-based models often overlook due to their inability to navigate non-Euclidean spatial features effectively (Han & Woo, 2022). Deng and Hooi (2021) propose a methodology that combines structure learning and GNN. This approach utilizes attention weights to enhance the explainability of

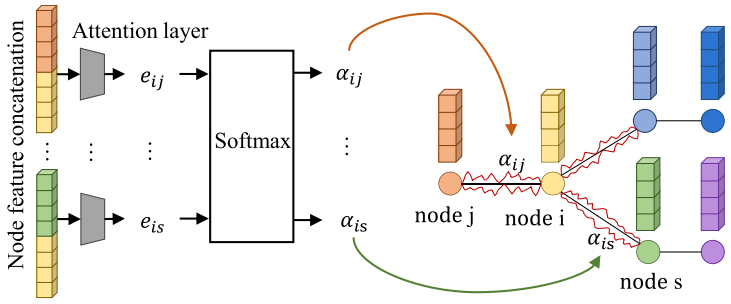

Fig. 1. The calculation of attention and the node feature update process. Attention mechanism calculation and node feature update procedure in GNN.

detected anomalies. The method accurately detects anomalies, captures sensor correlations effectively, and enables users to deduce the root cause of anomalies. Wang et al. (2021) proposed the Spatio-Temporal-Categorical Graph Neural Network (STC-GNN) to address multidimensional dynamic chain effects and enable fine-grained multi-incident co-prediction. This approach builds upon the insightful perspective of explicitly exploiting the underlying chainlike triggering mechanism to address the sparsity of rare incidents in a temporally fine-grained setting. Han and Woo (2022) propose FuSAGNet, a novel method that combines reconstruction and forecasting to capture relationships in multivariate time series. It integrates Sparse Autoencoder and Graph Neural Network, leveraging sparse latent representations for predicting future time series behavior. Additionally, recurrent feature embedding is used to learn graph structures and enhance relationship modeling within the data. Guo, Zhou, Zhao, and Gaaloul (2024) present EGNN, an efficient method for anomaly detection in IoT multivariate time series data. The method leverages SGA for correlation exploration among sensory data from diverse IoT devices through graph structure learning. the method provides a streamlined and precise solution for IoT applications, especially for rare anomalies. He et al. (2020) proposed TopoMAD, a robust stochastic seq2seq model for effectively modeling spatial and temporal dependencies in contaminated data. The model incorporates system topology information, sliding windows, and neural networks to extract features. Song et al. (2023) present Correlative-GNN, which incorporates Multi-Head Self-Attention and an Auto-Regression Ensemble Method for analyzing multivariate time series data in real-world operational clouds. The aforementioned studies have significantly advanced the domain of anomaly detection in high-dimensional time series.

GNN represent a paradigm shift in the handling of the structural intricacies of multivariate time series data. They offer two primary advantages that set them apart from traditional deep learning approaches.

- **Enhanced Structural Representation:** Traditional learning models typically perceive each time series as an isolated vector, failing to recognize the intricate structural relationships that exist between them. Graph Convolutional Networks (GCN) (Kipf & Welling, 2017) address this oversight by implementing convolution operations within a non-Euclidean framework. This approach allows for the integration of structural information by leveraging the connections (edges) between nodes (time series), thereby enriching the model's understanding of the data. Despite their considerable advancements, GCN continue to face challenges in accommodating asymmetric relationships and the varying significance of nodes within the graph. Graph Attention Networks (GAT) surmount these obstacles by introducing an attention mechanism that dynamically assesses node importance and adeptly manages asymmetric relationships. This innovation serves to reinforce the network's representational capabilities, while also facilitating its generalization across a range of disparate datasets.
- **Dynamic Feature Update and Propagation:** The core of graph learning networks lies in their ability to update and propagate internal features based on the graph's structure. The attention-based update method represents a significant leap forward, enabling a deeper exploration of structural data information. In the context of time series prediction, where each time series at a given timestamp is treated as a node feature, the graph structure dictates the mapping and propagation of these features. For instance, in this paper, a multivariate time series at timestamp t is denoted as $x = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(M)}]$, with M representing the number of nodes. The physical topology matrix, $A_{M\times M}$, encodes the connectivity between nodes, where $A_{ij} = 1$ signifies a connection between nodes i and j, and $A_{ij} = 0$ indicates no connection. The feature update process involves the transformation and aggregation of each node's features, with the attention calculation and node feature update process illustrated in Fig. 1.

While GAT has demonstrated their efficacy in accurately detecting anomalies across various real-world datasets, challenges remain, particularly in cloud computing networks where time dependency is critical for sensor embedding. The proposed EH-GAM-EGAN model seeks to bridge this gap by integrating GAT, thereby enhancing the model's ability to consider time dependencies and improve anomaly detection performance.

3. Problem formalization

This section represents the framework and foundational concepts underpinning the MTSAD task. This task is pivotal in numerous applications, ranging from industrial monitoring to financial fraud detection, where identifying aberrant patterns promptly can prevent potential losses or hazards. The notations and symbols utilized throughout this discourse are concisely cataloged in Table 1, which serves as a reference to aid in the comprehension of the ensuing discussion.

At the heart of MTSAD lies the challenge of processing and analyzing sequences of data points collected over time, each comprising multiple variables or features. Formally, we define a multivariate time series as $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T] = \mathbf{x}_{1:T}$, where each $\mathbf{x}_t \in \mathbb{R}^M$ represents the state of the system at time t, encapsulated by M distinct features. These features could, for instance, correspond to readings from a set of sensors in a manufacturing plant, financial indicators in stock market analysis, or any other multidimensional data collected sequentially over time.

To facilitate the training of anomaly detection models, the continuous time series $\mathbf{x}_{1:T}$ is segmented into N subsequences, each of length S, resulting in a training dataset $X \in \mathbb{R}^{N\times S}$. Each subsequence, denoted as $\mathbf{x}_{k,1:S}$ for $1 \le k \le N$, represents a window of S consecutive time points. This segmentation approach allows for the extraction of temporal patterns within each window, which are crucial for identifying anomalies.

In addition to the real data, our methodology incorporates synthetic data, represented by $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T]$, to simulate the presence of noise or non-anomalous deviations in the data. The elements of $\mathbf{z}$

Table 1
Notation description.

Notation	Description
M	Number of sensory devices/nodes.
N	Number of multivariate time series segments after segmentation.
S	Length of each multivariate time series segment after segmentation.
X	Training set of multivariate time series.
$x_{k,1:S}$	Denoting the kth segment of time series in X.
$\hat{\mathbf{y}}$	The output of MTSAD
P_R	Real data distribution.
P_G	Generated data distribution.

are drawn from a standard normal distribution, $z \sim P_z = \mathcal{N}(0,1)$, reflecting the assumption that normal operational data can exhibit random fluctuations that are not indicative of true anomalies.

The ultimate goal of MTSAD is to produce a predictive model capable of generating a binary output vector $\hat{\mathbf{y}} \in \mathbb{R}^T$, where each element $\hat{y}^{(t)} \in \{0,1\}$ signifies the model's assessment of whether the corresponding time point t is anomalous. Specifically, a value of $\hat{y}^{(t)} = 1$ flags the time point t as anomalous, thereby enabling timely intervention or further investigation.

4. Methodology

4.1. Overview of proposed model

The following section elucidates the architecture and operational dynamics of our novel anomaly detection framework, referred to as EH-GAM-EGAN. This framework is a confluence of prediction-based and reconstruction-based paradigms meticulously designed to elevate the efficacy of MTSAD. The EH-GAM-EGAN model comprises four essential modules, each playing a distinct role in the overall detection process. The synergy among these modules facilitates a comprehensive and nuanced approach to identifying anomalies. Below, we expound on the functionalities and interconnections of these modules.

- **Graph Attention-Based Temporal Prediction Module.** This module is central to our approach, utilizing Graph Neural Networks engineered to capture complex dependencies within the dataset. We deploy two layers of GNN to strike a balance between model depth and the risk of feature over-smoothing, which can obscure the distinctiveness of the features. To mitigate this risk, we incorporate attention mechanisms that refine the model's capability to extract nuanced features. This is critical for maintaining the integrity of temporal relationships and the dynamics of features across the time series.
- **LSTM-Based Temporal Prediction Module.** Working in conjunction with the Graph Attention-Based Module, this module consists of two LSTM layers followed by a linear layer, designed to effectively capture and model sequential patterns inherent in time series data. The dual training approach, integrating both the graph-based and LSTM-based models with the generative and discriminative components of the framework, ensures a comprehensive learning process attuned to both spatial and temporal characteristics of the data.
- **Enhanced Generative Adversarial Networks.** The generative adversarial component of our framework is depicted in the referenced architecture Fig. 2. It includes a generative module with three linear layers and a discriminative module also comprising three linear layers. Uniquely, the discriminator begins with a one-dimensional convolutional layer to extract complex features at a granular level from the time series data. During anomaly detection testing, the discriminator's error analysis is used to compute anomaly scores. Conventional GAN employing JS divergence might result in weight clipping that pushes weights towards the boundaries of the clipping range (Gulrajani, Ahmed, Arjovsky, Dumoulin, & Courville, 2017). We employ the Wasserstein distance with a gradient penalty (WGAN-GP) to overcome typical GAN issues like unstable training dynamics and mode collapse, ensuring stable model convergence and enhanced performance in anomaly detection tasks.
- **Anomaly Scoring.** The final phase of our model's operation involves calculating the anomaly detection score, which integrates scores from various model components: the reconstruction score produced by the generative module, the discrimination score generated by the discriminator, and the prediction scores obtained from the two prediction modules. This integrative scoring system maximizes the detection process's effectiveness, significantly minimizing false positives and enhancing the accuracy of anomaly detection.

The strategic amalgamation of advanced graph attention mechanisms, LSTM-based temporal modeling, and robust generative adversarial networks within the EH-GAM-EGAN framework enables superior performance in MTSAD. This methodology can ensure a nuanced and thorough approach to anomaly detection, setting a new standard in the field. We will provide more information about the design specifics of the four modules.

4.2. Graph attention-based temporal prediction module

The capacity to comprehend and interpret the intricate inter-depen dencies among various data points is essential for effective anomaly detection in multivariate time series data. To enhance the detection, we have integrated a graph attention-based feature extractor into our framework. This module primarily utilizes GAT to highlight the significance and influence of individual nodes based on the structural context of the data. GAT apply attention mechanisms to calculate the weights of edges connecting nodes, thereby identifying key contributors to the overall structure and prediction outcomes. As illustrated in Fig. 3, our GAT architecture processes data through multiple layers, each refining the abstraction of features extracted from the previous one. This sequential processing progressively boosts the model's capability to discern and synthesize complex patterns and dependencies.

In the context of time series analysis, the continuous time series $\mathbf{x}_{1:T}$ is divided into N subsequences, each of length S, to form a training dataset $X \in \mathbb{R}^{N\times S}$. Each subsequence, denoted as $\mathbf{x}_{k,1:S}$ for $1 \le k \le N$, constitutes a window of S consecutive time points. For instance, within subsequence k, the input to our model at a specific time t is defined by a sliding window of size w_{in}, which captures a segment of the historical data. This is represented mathematically as:

$$\mathbf{h}_{k,t} = \left[\mathbf{x}_{k,t-w_{in}}, \mathbf{x}_{k,t-w_{in}+1}, \ldots, \mathbf{x}_{k,t-1}\right] \tag{1}$$

Here, $\mathbf{h}_{k,t} \in \mathbb{R}^{M\times w_{in}}$ serves as the input matrix for the sliding window at time t, encapsulating the temporal dynamics of the multivariate time series. The multivariate time series point at time t is represented by $\mathbf{x}_{k,t} = [x_{k,t}^{(1)}, x_{k,t}^{(2)}, \ldots, x_{k,t}^{(M)}]$, which denotes the values at M nodes. The physical topology matrix $A_{M\times M}$ indicates the connectivity between nodes i and j, with $A_{ij} = 1$ indicating a connection and $A_{ij} = 0$ signifying the lack thereof. This research models the physical topology as a strongly connected graph.

To compute the combined representation $\mathbf{z}^{(i)}$ of node i at time t, we employ the following formula:

$$\mathbf{z}_{k,t}^{(i)} = \mathrm{ReLU}\left(\alpha_{i,i}\mathbf{W}\mathbf{h}_{k,t}^{(i)} + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{W}\mathbf{h}_{k,t}^{(j)}\right) \tag{2}$$

in this equation, $\mathbf{h}_{k,t}^{(i)} \in \mathbb{R}^{w_{in}}$ represents the input feature of node i, and $\mathcal{N}(i) = \{j | A_{ji} > 0\}$ denotes the set of adjacent nodes to node i. The matrix $\mathbf{W} \in \mathbb{R}^{M\times w_{in}}$ is a trainable weight matrix that applies a linear transformation to each node's features. The attention coefficients $\alpha_{i,j}$, which quantify the importance of node j's features to node i, are

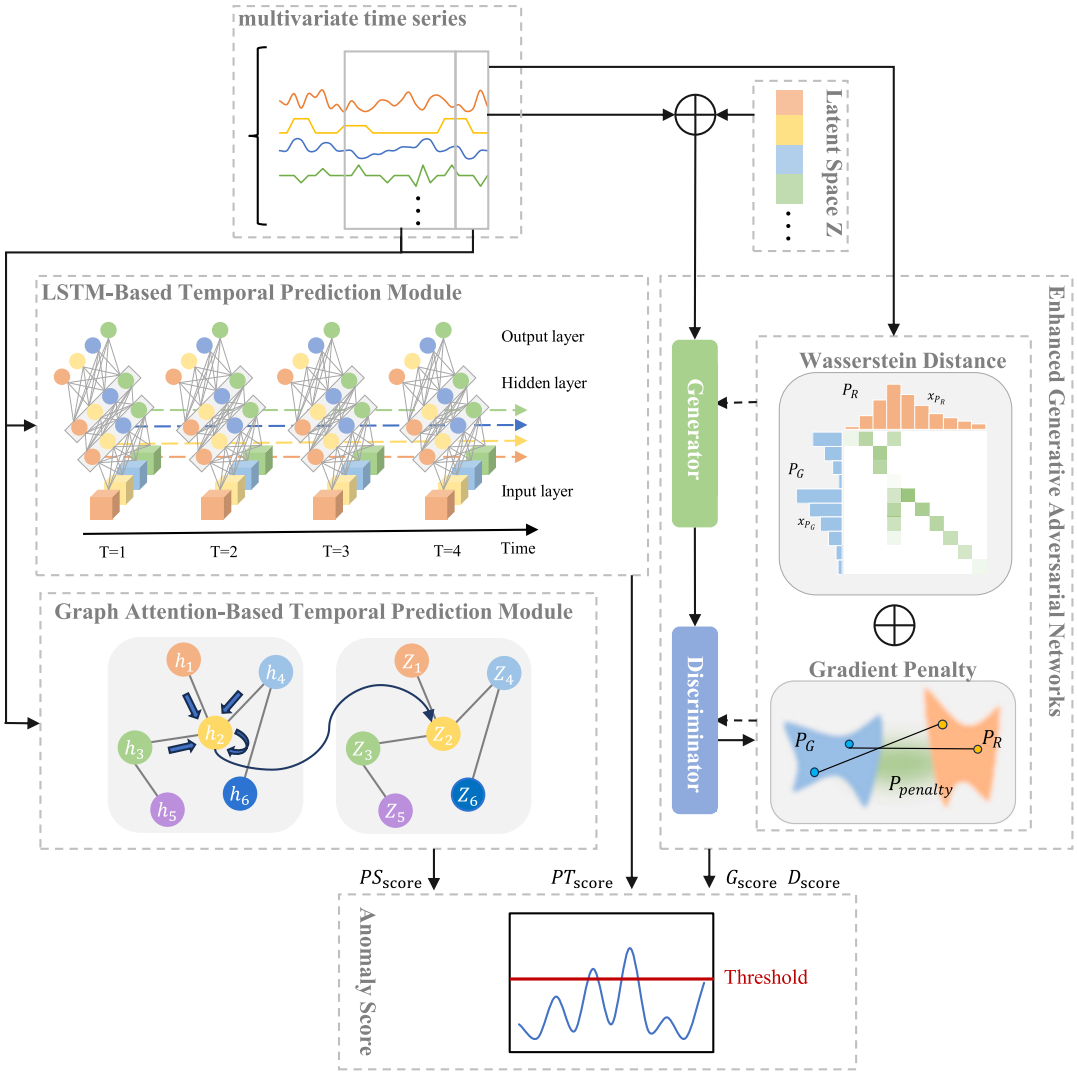

Fig. 2. The architecture of EH-GAM-EGAN for MTSAD.

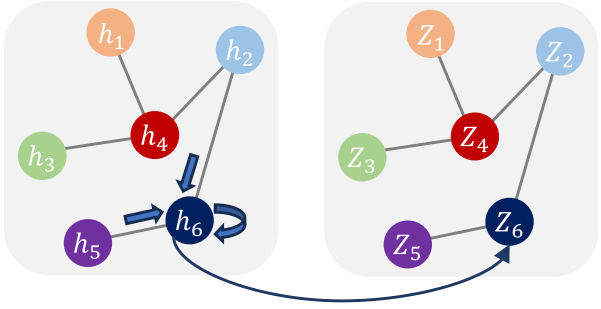

Fig. 3. Graph attention-based feature extractor.

computed as follows:

$$\pi(i,j) = \text{LeakyReLU}\left(\mathbf{a}^{\top}\left(W\mathbf{h}_{k,t}^{(i)} \oplus W\mathbf{h}_{k,t}^{(j)}\right)\right) \tag{3}$$

$$\alpha_{i,j} = \frac{\exp(\pi(i,j))}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp(\pi(i,k))} \tag{4}$$

here, $\mathbf{a}$ is a vector of learned coefficients that facilitate the computation of attention scores, and $\oplus$ denotes the concatenation operation. The LeakyReLU function introduces non-linearity in the computation of attention scores, while the softmax function ensures that the attention coefficients across all nodes sum up to 1, thus normalizing the influence of each node.

Following the extraction of node representations through the graph attention-based feature extractor, we obtain a comprehensive set of features $\left\{\mathbf{z}_{k,t}^{(1)}, \mathbf{z}_{k,t}^{(2)}, \ldots, \mathbf{z}_{k,t}^{(M)}\right\}$ for all nodes within the time series data. These features are then utilized to predict the future state of the time series, denoted as $\hat{\mathbf{x}}_{k,t} = \left\{\mathbf{z}_{k,t}^{(1)}, \mathbf{z}_{k,t}^{(2)}, \ldots, \mathbf{z}_{k,t}^{(M)}\right\}$. This predicted output $\hat{\mathbf{x}}_{k,t}$ is a synthesis of the learned representations, aiming to closely approximate the actual observed data $\mathbf{x}_{k,t}$ at time t.

To quantify the accuracy of our predictions and inform the optimization of our model, we employ the mean squared error (MSE) as the objective function. The MSE is a widely recognized measure for evaluating regression model performance, offering a straightforward measure of the difference between predicted and actual values, denoted

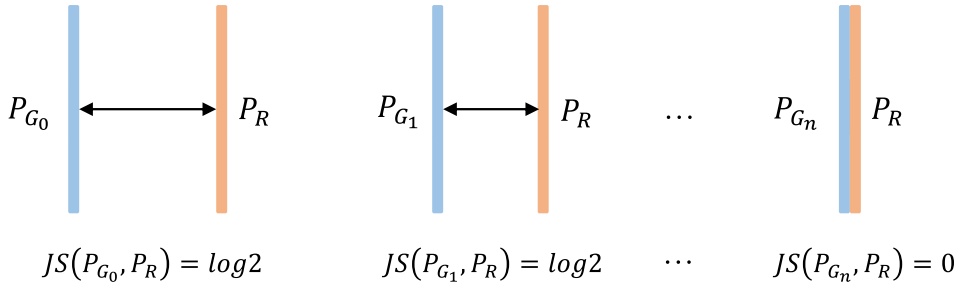

Fig. 4. JS divergence is always log2 if two distributions do not overlap.

as L_{PS}, is articulated as follows:

$$L_{PS} = \frac{1}{N}\sum_{k=1}^{N} \| \mathbf{x}_{k,t} - \hat{\mathbf{x}}_{k,t} \|^2 \tag{5}$$

where N denotes the total count of samples in the dataset, and $\| \cdot \|^2$ denotes the squared Euclidean norm, which calculates the sum of the squared differences between the predicted and observed values across all dimensions of the time series data. The objective function L_{PS} thus encapsulates the average squared discrepancy across all samples, offering a thorough assessment of the model's predictive accuracy.

4.3. LSTM-based temporal prediction module

This section introduces an LSTM-Based Temporal Prediction Module, a sophisticated component designed to process multivariate time series data and generate precise future predictions. This module is structured around two LSTM layers followed by a linear fully connected layer, catering to the intricate dynamics of time series data. The choice of LSTM networks is motivated by their proven efficacy in capturing long-term dependencies and patterns within sequential data, making them ideally suited for time series forecasting.

Given an input time series $\mathbf{x}_{k,1:S}, k = 1, \ldots, N$, where S denotes the sequence length and N the number of samples, our prediction model, denoted as $\mathrm{PT}(\cdot;\eta)$, is parameterized by η. The model is tasked with forecasting a target window of length w_{out}, based on an input window of length w_{in}. In other words, we have $PT(\mathbf{x}_{k,1:t_0};\eta) = \hat{\mathbf{x}}_{k,t_0+1:t_0+w_{out}}$. Specifically, the model aims to predict the values for the time steps $[t_0+1, t_0+w_{out}]$ utilizing the historical data $\mathbf{x}_{k,1:t_0}$, where $t_0 + w_{out} = S$. This predictive task delineates two distinct temporal ranges: the conditional range $[1, t_0]$, which provides the historical context, and the target range $[t_0+1, S]$, which specifies the future time steps to be forecasted.

To evaluate the performance of the LSTM-Based Temporal Prediction Module and guide its training, we also employ the MSE loss as the objective function. It quantifies the discrepancy between the predicted values $\hat{\mathbf{x}}_{k,t_0+1:S}$ and the actual observed values $\mathbf{x}_{k,t_0+1:S}$ over the target range. The objective function for the prediction module, L_{PT}, is defined as follows:

$$L_{PT} = \frac{1}{N}\sum_{k=1}^{N} \| \mathbf{x}_{k,t_0+1:S} - \hat{\mathbf{x}}_{k,t_0+1:S} \|^2 \tag{6}$$

this module represents a cornerstone of our anomaly detection framework, leveraging the strengths of LSTM networks to provide reliable and accurate forecasts.

4.4. Enhanced generative adversarial networks

4.4.1. Generative adversarial networks

GAN represents a groundbreaking framework in machine learning, particularly in generative modeling. The GAN architecture is predicated on an adversarial process, wherein two distinct models are trained concurrently: a Generator G and a Discriminator D. The Generator G endeavors to learn the underlying data distribution of the training dataset, while the Discriminator D aims to discern whether a given instance is drawn from the actual training data or produced by the generative model. This adversarial dynamic fosters a competitive environment, compelling the generative model to produce increasingly realistic data, with the ultimate goal of deceiving the discriminative model into misclassification. The essence of the GAN framework is encapsulated in a mini-max game between the generative and discriminative models, governed by the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_R}[\log D(x)] + \mathbb{E}_{\tilde{x}\sim P_G}[\log(1 - D(\tilde{x}))] \tag{7}$$

where, P_R represents the distribution of real data, and P_G denotes the distribution of data generated by the model, where $\tilde{x} = G(z)$ and $z \sim p(z)$ is a latent variable sampled from a predefined distribution (e.g., uniform or normal distribution). $\min_G$ indicates the objective that is to minimize the function with respect to the Generator G, and $\max_D$ indicates the objective that is to maximize the function concerning the Discriminator D. $V(D,G)$ is the value function between the Discriminator D and the Generator G. $\mathbb{E}_{x\sim P_R}[\log D(x)]$ is the expected log of the Discriminator output $D(x)$ when x is sampled from the real data distribution P_R. $\mathbb{E}_{\tilde{x}\sim P_G}[\log(1 - D(\tilde{x}))]$ is the expected log of 1 minus the Discriminator output $D(\tilde{x})$ when $\tilde{x}$ is sampled from the generated data distribution P_G. $\tilde{x} = G(z)$ means the Generator G maps the latent variable z, sampled from a predefined distribution (e.g., uniform or normal), to the generated sample $\tilde{x}$.

The Discriminator D is trained to maximize the objective function, effectively improving its ability to distinguish between real and generated data. Conversely, the Generator G is trained to minimize the same objective function, striving to produce data that closely mimics the real data distribution.

A critical challenge in the traditional GAN framework is the reliance on minimizing the Jensen–Shannon (JS) divergence to approximate the distance between the real data distribution P_R and the generated data distribution P_G. This approach can lead to the issue of gradient vanishing, particularly when the support of the two distributions does not overlap, as illustrated in Fig. 4. The JS divergence remains constant at $log2$ in scenarios where there is no overlap between the distributions, hindering the training process and potentially leading to significant discrepancies between the real and generated data distributions.

The adversarial training process of GAN, characterized by this mini-max game, lays the foundation for a wide array of applications in generative modeling. By iteratively refining the capabilities of both the generative and discriminative models, GAN facilitate the generation of highly realistic data, contributing to advancements in fields of Anomaly detection.

4.4.2. Wasserstein GAN

The introduction of Wasserstein Generative Adversarial Networks (WGAN) by Arjovsky et al. marks a significant advancement in addressing the challenges of gradient vanishing and mode collapse that are prevalent in traditional GAN. The WGAN framework modifies the conventional GAN architecture by redefining the role of the discriminator

to estimate the Wasserstein distance between the real data distribution P_R and the generated data distribution P_G. This shift from the traditional discriminator to a critic that computes the Wasserstein distance introduces a more stable gradient flow, facilitating more effective training of the generative model.

The Wasserstein distance, unlike the Kullback–Leibler (KL) divergence or the Jensen–Shannon (JS) divergence, provides a continuous measure of distance between distributions, even when they are disjoint. This property is crucial for the learning process, as it ensures a smoother and more stable gradient descent, thereby overcoming the limitations associated with abrupt changes or infinite values in divergence measures. The Wasserstein distance's ability to reflect the closeness of distributions, regardless of their overlap, is illustrated in Fig. 5 and further elaborated in Fig. 6, showcasing its superiority in adversarial training contexts.

The Wasserstein distance is fundamentally based on optimal transport theory, which aims to identify the most efficient method for transforming one distribution into another. This is represented by a transport plan matrix $\nu \in \mathcal{R}^{I\times J}$, where each element signifies the mass transferred from the generated data distribution P_G to the real data distribution P_R. The intensity of each element in ν correlates with the volume of mass relocated, offering a visual representation of the transport process, as illustrated in Fig. 5. The computation of the Wasserstein distance involves identifying the optimal transport plan ν that minimizes the total cost of transformation. This is mathematically expressed as:

$$B(\nu) = \sum_{x_{P_G}, x_{P_R}} \nu(x_{P_G}, x_{P_R}) \| x_{P_G} - x_{P_R} \| \tag{8}$$

where $B(\nu)$ denotes the average distance or cost associated with a given transport plan ν. The Wasserstein distance $W(\cdot)$ is then defined as the minimum of these costs across all possible transport plans Π, as shown in the equation:

$$W(P_G, P_R) = \min_{\upsilon \in \Pi} B(\upsilon) \tag{9}$$

here, Π represents the set of all conceivable joint distributions between P_G and P_R, with $\upsilon \in \Pi$ specifying a particular transport plan. This formulation underscores the Wasserstein distance's ability to provide a continuous and robust metric for quantifying the dissimilarity between P_G and P_R, even in scenarios where the two distributions do not overlap.

The WGAN framework leverages the Wasserstein distance to redefine the adversarial training process, enhancing stability and mitigating the risk of mode collapse. The objective function of WGAN is articulated as follows:

$$\min_G \max_{D\in D} \mathbb{E}_{\boldsymbol{x}\sim P_R}[D(\boldsymbol{x})] - \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim P_G}[D(\widetilde{\boldsymbol{x}})] \tag{10}$$

here, $\min_G$ means the objective that is to minimize the function for the Generator G. $\max_{D\in D}$ denotes the objective that is to maximize the function concerning the Discriminator D, where D belongs to the set D of 1-Lipschitz continuous functions. $\mathbb{E}_{\boldsymbol{x}\sim P_R}[D(\boldsymbol{x})]$ is the expected value of the Discriminator output $D(\boldsymbol{x})$ when $\boldsymbol{x}$ is sampled from the real data distribution P_R. Similarly, $\mathbb{E}_{\widetilde{\boldsymbol{x}}\sim P_G}[D(\widetilde{\boldsymbol{x}})]$ is the expected value of the Discriminator output $D(\widetilde{\boldsymbol{x}})$ when $\widetilde{\boldsymbol{x}}$ is sampled from the generated data distribution P_G. The key difference from the standard GAN objective is the use of the Wasserstein distance, which is defined as the maximum difference between the expected Discriminator outputs on real and generated samples. This formulation encourages the Discriminator to behave as a 1-Lipschitz function, thereby stabilizing the training process and reducing the risk of mode collapse.

In the development and optimization of WGAN, a critical methodological innovation is the enforcement of the Lipschitz constraint on the discriminator model, denoted as D. This constraint is pivotal for the theoretical underpinnings of WGAN, ensuring that the discriminator function remains 1-Lipschitz continuous. A practical approach to maintaining this constraint involves clipping the weights of the discriminator model to a predefined interval, specifically $[-c, c]$. This technique, known as weight clipping, serves to limit the magnitude of weight adjustments during the training process, thereby preserving the Lipschitz continuity of the discriminator model. However, the ambiguity of the hyperparameter c directly affects the range of weight clipping, impacting the numerical computations during the model training process, and potentially leading to unstable training or less than ideal results.

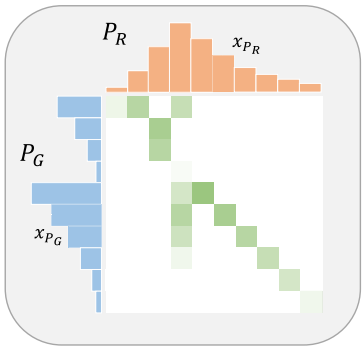

Fig. 5. Wasserstein distance between two distributions.

4.4.3. WGAN-gradient penalty

In the evolution of WGAN, a significant methodological enhancement was introduced with the WGAN-GP model. This advancement addresses the limitations associated with the weight clipping strategy in the discriminator model, a method previously employed to enforce the Lipschitz constraint. The WGAN-GP model innovates by incorporating a gradient penalty (GP) term into the objective function, which serves to regulate the gradient norm of the discriminator model, ensuring it remains proximate to a predefined value, typically 1. This section elucidates the formulation and implications of the gradient penalty approach.

The objective function used in WGAN-GP includes a gradient penalty term, which is designed to mitigate the issues of gradient disappearance and parameter divergence that may arise from the weight clipping strategy. The revised objective function is expressed as follows:

$$L = \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim P_G}[D(\widetilde{\boldsymbol{x}})] - \mathbb{E}_{\boldsymbol{x}\sim P_R}[D(\boldsymbol{x})] + \lambda \mathbb{E}_{\overline{\boldsymbol{x}}\sim P_{\overline{\boldsymbol{x}}}}\left[\left(\|\nabla_{\overline{\boldsymbol{x}}} D(\overline{\boldsymbol{x}})\|_2 - 1\right)^2\right] \tag{11}$$

in this formulation, $\overline{\boldsymbol{x}} = \epsilon \boldsymbol{x} + (1-\epsilon)\widetilde{\boldsymbol{x}}$ represents interpolated samples between real data $\boldsymbol{x}$ and generated data $\widetilde{\boldsymbol{x}}$, with ϵ being a random variable sampled uniformly from the interval $[0, 1]$. $\|\nabla_{\overline{\boldsymbol{x}}} D(\overline{\boldsymbol{x}})\|_2$ is the L2 norm of the gradient of the Discriminator output $D(\overline{\boldsymbol{x}})$ with respect to the interpolated sample $\overline{\boldsymbol{x}}$. The gradient penalty coefficient λ plays a crucial role in determining the degree of constraint enforcement, directly influencing the balance between model flexibility and adherence to the Lipschitz condition. The gradient penalty term $\lambda \mathbb{E}_{\overline{\boldsymbol{x}}\sim P_{\overline{\boldsymbol{x}}}}\left[\left(\|\nabla_{\overline{\boldsymbol{x}}} D(\overline{\boldsymbol{x}})\|_2 - 1\right)^2\right]$ serves to regularize the Discriminator by ensuring that its gradient norm is close to 1, thereby enforcing the Lipschitz constraint without the need for weight clipping.

The GP mechanism is predicated on the observation that enforcing a unit gradient norm constraint across the entire discriminator function is impractical. Instead, GP introduces a more nuanced approach that targets specific regions within the data space. This is achieved by defining the sampling of an interpolated distribution $P_{\overline{x}}$, which is constructed along lines connecting pairs of points. These points are drawn uniformly from the real data distribution P_R and the generator model distribution P_G. Fig. 7 visually encapsulates this concept, illustrating the strategic sampling process facilitated by the gradient penalty.

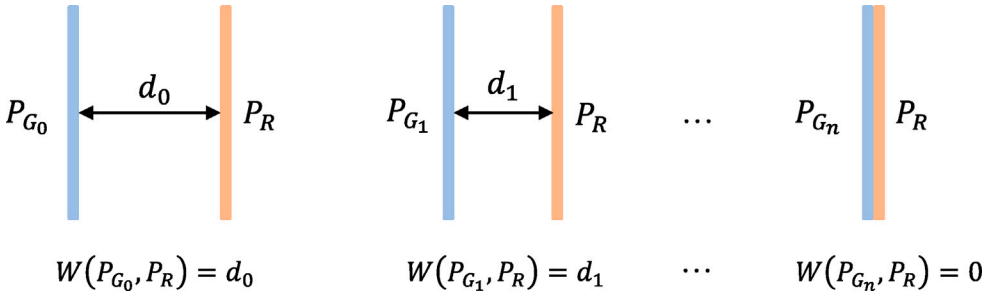

Fig. 6. Wasserstein distance between the generated and the real distributions across adversarial training epochs. Here, n represents the epoch count.

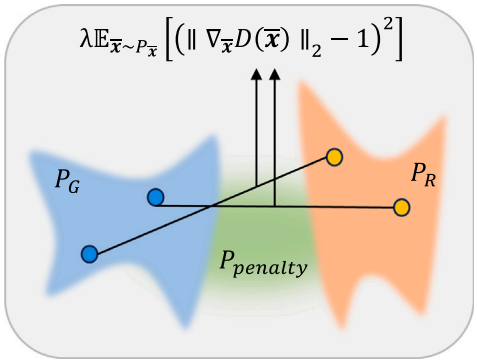

Fig. 7. The area of the gradient penalty constraint between P_G and P_R.

4.5. Synchronous multi-model fusion training for EH-GAM-EGAN

This section examines the methodology employed in simultaneous training, with a particular focus on the interaction between the generator and discriminator models within the context of time series generation. The generator, denoted as $G(\cdot;\theta)$, where θ represents the model parameters, is tasked with reconstructing time series values over a specified interval. Given an input time series $x_{k,1:S}$ where $k = 1, \ldots, N$, the generator is tasked with reconstructing the values for time steps $[t_0+1, t_0+\tau]$, where $t_0 + \tau = S$. This is done based on the condition of $\mathbf{x}_{k,t_0+1:S}$ and an added noise vector z. The number of time steps, denoted by τ, that the generator is trained to reconstruct is a critical parameter in this process. The reconstructed time series is represented as $\hat{x}_{k,t_0+1:S} = G(\mathbf{x}_{k,t_0+1:S} + \mathbf{z}; \theta)$.

The discriminator, designated as $D(\cdot;\omega)$, where ω denotes the model parameters, evaluates the similarity between the output generated by the model and the actual time series data. The objective function for the discriminator is formulated as follows:

$$L_D = \frac{1}{N}\sum_{k=1}^{N} D(\hat{\mathbf{x}}_{k,t_0+1:S}) - \frac{1}{N}\sum_{k=1}^{N} D(\mathbf{x}_{k,t_0+1:S}) + \lambda \frac{1}{N}\sum_{k=1}^{N}\left(\left(\left\|\nabla_{\bar{x}_k} D(\bar{x}_k)\right\|_2 - 1\right)^2\right) \tag{12}$$

in this equation, the mean of the time series at time point k is given by: $\bar{\mathbf{x}}_k = \epsilon \mathbf{x}_{k,t_0+1:S} + (1-\epsilon)\hat{\mathbf{x}}_{k,t_0+1:S}$ where ϵ is a random variable sampled uniformly from the interval [0,1]. This formulation incorporates a GP term to enforce the Lipschitz constraint, ensuring the stability of the training process. The objective function for the generator model aims to maximize the discriminator's error in distinguishing between real and generated time series data. It is defined as follows:

$$L_G = -\frac{1}{N}\sum_{k=1}^{N} D(\hat{\mathbf{x}}_{k,t_0+1:S}) \tag{13}$$

this function motivates the generator to create outputs that are indistinguishable from real-time series data, as determined by the discriminator.

The synchronous multi-model training of EH-GAM-EGAN, as depicted in Algorithm 1, commences with the initialization of the parameters for LSTM-based temporal and graph attention-based spatial prediction models, in addition to those for the discriminator and generator. These models are key components designed to interact and improve through mutual feedback in a tightly coupled training loop. Training proceeds iteratively until the generator's parameters, denoted as θ_0, converge. Each main iteration comprises multiple sub-iterations, corresponding to the number of discriminator updates per generator update, set by $n_{(critic)}$. Within each sub-iteration, the following steps occur: The following steps are then carried out:

- Data Sampling and Prediction (Lines 4–7);
- Loss Computation (Line 8);
- Parameter Updates (Lines 10–13).

Following these discriminator and predictor updates, the generator is then refined through a similar process. A new batch of generator outputs is sampled, and the loss $L_G^{(i)}$ is computed for the generator. This facilitates the update of the generator's parameters via the Adam optimizer. The Adam optimizer utilizes not only the loss derived from the generator's output but also gradients from the discriminator's recent updates. This ensures that the generator's parameters are updated in a way that maintains adversarial learning.

This training protocol ensures that all components (the temporal and spatial predictors, the discriminator, and the generator) are not only updated independently based on their specific loss functions but also influenced by the ongoing learning in other parts of the model. This synchronous update mechanism helps to harmonize the learning across different model components, thereby potentially leading to a more robust and generalized model performance. The application of the Adam optimizer, as detailed in the training steps, employs gradient penalties to regulate the training dynamics. This addresses the impact of rapidly changing loss landscapes and ensures stable convergence.

The time complexity of the Synchronous Multi-Model Training algorithm can be decomposed into several components, each corresponding to a specific operation within the training loop. The primary operations include sampling data, computing losses, and updating model parameters. The analysis assumes that the time complexity of each operation is primarily influenced by the batch size m, the number of discriminator iterations per generator iteration n_{critic}, and the inherent complexity of the models involved.

- **Sampling Operations:** The algorithm conducts a series of sampling operations during each iteration. These operations involve sampling real data and corresponding latent variables, generating outputs from the graph attention-based spatial predictor and the LSTM-based temporal predictor, and creating synthetic data using

Algorithm 1 Synchronous Multi-Model Training of EH-GAM-EGAN.

Require: Gradient penalty coefficient λ, number of discriminator iterations per generator iteration n_{critic}, batch size m, Adam optimizer hyperparameters ϕ_1,ϕ_2,ψ_1, and ψ_2.
Require: Initial parameters for graph attention-based temporal predictor a_0, W_0, initial parameters for LSTM-based temporal predictor η_0, initial parameters for discriminator ω_0, initial parameters for generator θ_0.
while θ_0 has not converged **do**
  for $t = 1, \dots, n_{critic}$ **do**
    for $i = 1, \dots, m$ **do**
      Sample real data and corresponding latent variable;
      Obtain prediction output $PS(\mathbf{x}_{k,1:t_0})$ from the graph attention-based temporal predictor;
      Obtain prediction output $PT(\mathbf{x}_{k,1:t_0})$ from the LSTM-based temporal predictor;
      Generate synthetic data $G(\mathbf{x}_{k,t_0+1:S} + \mathbf{z})$ using the generator;
      Compute losses $L_{\mathrm{PT}}^{(i)}$ for temporal predictions (Eq. (5)), $L_{\mathrm{PS}}^{(i)}$ for spatial predictions (Eq. (6)), and $L_D^{(i)}$ for discrimination (Eq. (12)).
    end for
    Update spatial predictor parameter $a \leftarrow \mathrm{Adam}\left(\nabla_a \frac{1}{m}\sum_{i=1}^{m} L_{PS}^{(i)}, a, \phi_1\right)$;
    Update spatial predictor parameter $W \leftarrow \mathrm{Adam}\left(\nabla_W \frac{1}{m}\sum_{i=1}^{m} L_{PS}^{(i)}, W, \phi_1\right)$;
    Update temporal predictor parameter $\eta \leftarrow \mathrm{Adam}\left(\nabla_\eta \frac{1}{m}\sum_{i=1}^{m} L_{PT}^{(i)}, \eta, \phi_2\right)$;
    Update discriminator parameter $\omega \leftarrow \mathrm{Adam}\left(\nabla_\omega \frac{1}{m}\sum_{i=1}^{m} L_{D}^{(i)}, \omega, \phi_1, \phi_2, \psi_1, \psi_2\right)$;
  end for
  Sample a new batch of generator output;
  Compute generator loss $L_G^{(i)}$ (Eq. (13)) ;
  Update generator parameter $\theta \leftarrow \mathrm{Adam}\left(\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} L_{G}^{(i)}, \theta, \phi_1, \phi_2, \psi_1, \psi_2, \omega,\right)$
  return η,a,W,ω,θ
end while

the generator. The time complexity associated with these operations is generally linear with respect to the batch size m, which is denoted as $\mathcal{O}(m)$. However, the complexity may vary depending on the specific implementation and the efficiency of the sampling procedures.

- **Loss Computation:** The computation of losses for temporal predictions ($L_{\mathrm{PT}}^{(i)}$), spatial predictions ($L_{\mathrm{PS}}^{(i)}$), and discrimination ($L_D^{(i)}$) represents a critical step within each iteration. The complexity of loss computation is influenced by the batch size m and the computational complexity of the models' forward passes. Assuming that the forward pass of each model has a complexity of $\mathcal{O}(f)$, where f represents the operations involved in the forward computation, the total complexity for loss computation across all samples in the batch is $\mathcal{O}(m \cdot f)$.
- **Parameter Updates:** The parameters of the temporal predictor, spatial predictor, and discriminator are updated using the Adam optimizer, which has a linear complexity with respect to the number of parameters being updated. If p represents the total number of parameters across all models, the complexity of the parameter update step is $\mathcal{O}(p)$. It is important to note that this step is repeated n_{critic} times for the discriminator updates within each generator iteration.

The overall time complexity of the Synchronous Multi-Model Training algorithm per generator iteration can be approximated as follows: $O(n_{critic} \cdot m \cdot f + n_{critic} \cdot p + m \cdot f + p)$ This expression accounts for the discriminator updates, loss computations, and parameter updates for all models involved in the training process.

4.6. Anomaly scoring

This section outlines the methodology employed for anomaly scoring, a crucial phase in which the model assesses test set data to ascertain anomaly scores. The approach is inspired by the framework introduced in MADGAN (Li et al., 2019), which integrates multiple error dimensions to formulate a comprehensive anomaly score (AD_{score}). The score incorporates the reconstruction error (G_{score}), the discrimination error (D_{score}), and the prediction errors from both the graph attention-based temporal prediction module (PS_{score}) and the LSTM-based temporal prediction module (PT_{score}). The formulation of these scores is of paramount importance for distinguishing normal instances from anomalies within the dataset.

The anomaly scoring mechanism is based on the aggregation of errors from various modules, each of which contributes a unique perspective on the data's normalcy or deviation therefrom. The scores are computed as follows:

- The PS_{score} is derived by evaluating the squared Euclidean distance between the actual time series data $\mathbf{x}_{k,t_0+1:S}$ and the predictions made by the graph attention-based temporal prediction module $PS(\mathbf{x}_{k,1:t_0})$, summed across all instances N:
$$PS_{\mathrm{score}} = \sum_{k=1}^{N} \| \mathbf{x}_{k,t_0+1:S} - PS(\mathbf{x}_{k,1:t_0}) \|^2 \tag{14}$$
- Similarly, the PT_{score} quantifies the prediction error from the LSTM-based temporal prediction module, employing the same squared Euclidean distance metric:
$$PT_{\mathrm{score}} = \sum_{k=1}^{N} \| \mathbf{x}_{k,t_0+1:S} - PT(\mathbf{x}_{k,1:t_0}) \|^2 \tag{15}$$
- The G_{score} reflects the reconstruction error, measuring the discrepancy between the original time series data and the reconstructed data generated by the generator module G, again using the squared Euclidean distance:
$$G_{\mathrm{score}} = \sum_{k=1}^{N} \| \mathbf{x}_{k,t_0+1:S} - G(\mathbf{x}_{k,t_0+1:S} + \mathbf{z}) \|^2 \tag{16}$$
- The D_{score} captures the discrimination error, calculated as the sum of the deviations of the discriminator module's outputs from unity, across all instances:
$$D_{\mathrm{score}} = \sum_{k=1}^{N} \left(1 - D\left(G(\mathbf{x}_{k,t_0+1:S} + \mathbf{z})\right)\right) \tag{17}$$

The composite anomaly score, AD_{score}, is then formulated by linearly combining these individual scores, weighted by coefficients α, β, γ, and δ:
$$AD_{\mathrm{score}} = \alpha PS_{\mathrm{score}} + \beta PT_{\mathrm{score}} + \gamma G_{\mathrm{score}} + \delta D_{\mathrm{score}} \tag{18}$$
where the sum of the weights equals one ($\alpha + \beta + \gamma + \delta = 1$). These weights can be adjusted based on empirical evidence, allowing for the fine-tuning of the model's sensitivity to different types of errors.

The anomaly detection phase employs the AD_{score} to identify outliers within the dataset. This is achieved by establishing a threshold, beyond which a data point is considered anomalous. The determination of this threshold is flexible, accommodating various statistical or machine learning-based approaches to best suit the dataset's characteristics. Once a data point's AD_{score} exceeds this threshold, it is flagged as an outlier, signaling a deviation from the expected pattern of the time series.

Table 2
Statistical overview of the experimental datasets.

Dataset	Train	Test	Dimensions	Anomaly rate (%)
SMD	708,405	708,420	38	4
PSM	132,481	26,497	25	28
MSL	58,317	73,729	55	11
SMAP	135,183	427,617	25	13

5. Experiments

In this section, we will begin with a detailed description of the experimental datasets and performance metrics utilized in this study. Following this, we will conduct several experiments to illustrate the effectiveness of our proposed model.

5.1. Datasets

To ensure a thorough and precise evaluation of the model's capabilities, we engaged four distinct datasets, each derived from operational Internet of Things (IoT) systems in the real world. These datasets, which were carefully selected for their relevance and complexity, encompass a range of domains, including server systems and aerospace systems. These domains present unique challenges and demands for precise anomaly detection, making them ideal test cases for the model under investigation. The datasets employed are detailed in Table 2, offering a comprehensive view of their statistical properties and domain-specific characteristics.

- **SMD (Server Machine Dataset)** (Su et al., 2019) is a dataset originating from 28 active servers in a cloud platform that encompasses real-time server metrics. Anomalies within this test set are meticulously labeled by domain experts using event reports in order to ensure both accuracy and relevance.
- **PSM (Pooled Server Metrics)** (Abdulaal, Liu, & Lancewicki, 2021) is a comprehensive aggregation of 26-dimensional sensory data pertaining to numerous application server nodes operated by the online auction and shopping marketplace eBay. The dataset encapsulates a vast array of diverse server metrics, reflecting the multifaceted nature of effective server system monitoring.
- **MSL (Mars Science Laboratory ROVER)** (Hundman, Constantinou, Laporte, Colwell, & Soderstrom, 2018) is a dataset furnished by NASA. It contains 55-dimensional telemetry data, which includes anomaly data identified through Incident Surprise Anomaly (ISA) reports. This data offers insights into challenges faced by spacecraft monitoring systems.
- **SMAP (Soil Moisture Active Passive satellite)** dataset (Hundman et al., 2018) comprises 25-dimensional telemetry data. Similar to MSL, It includes anomalous data derived from ISA reports, reflecting the intricacies of satellite monitoring systems.

A subset of data points was selected from each dataset's training and test sets to facilitate the experiments. In particular, 50,000 data points were chosen from SMD, while 20,000 data points were extracted from the training and test sets of PSM, MSL, and SMAP. The selection process was guided by maintaining a balance between comprehensiveness and manageability, ensuring a thorough evaluation of the EH-GAM-EGAN model across various scenarios.

The statistical details of the datasets reveal significant diversity in terms of dimensionality and anomaly rates. For instance, the SMD dataset, with its 38 dimensions and a relatively low anomaly rate of 4%, presents a different set of challenges compared to the PSM dataset, which has a 28% anomaly rate within its 25 dimensions. Similarly, the MSL and SMAP datasets, with their high dimensionality and moderate anomaly rates, offer unique insights into the performance of anomaly detection models in complex aerospace systems.

5.2. Evaluation metrics

Anomaly detection faces the challenge of class imbalance, where the number of normal instances significantly exceeds that of anomalies. This imbalance necessitates the use of evaluation metrics that can accurately reflect the performance of anomaly detection methods, especially when distinguishing relatively rare anomalous instances. To address this, we utilize a set of standard evaluation metrics: precision (Pre), recall (Rec), and the F1 score (F1). These metrics were selected for their effectiveness in offering a balanced assessment of a model's performance, particularly in scenarios characterized by class imbalances. The formulas for each of these metrics can be found below.

$$\mathrm{Pre} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{19}$$

$$\mathrm{Rec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{20}$$

$$\mathrm{F1} = 2 \times \frac{\mathrm{Pre} \times \mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}} \tag{21}$$

in these equations, TP (True Positives) represents the number of anomalies accurately identified by the model, FP (False Positives) refers to the number of normal instances incorrectly labeled as anomalies, and FN (False Negatives) indicates the number of anomalies that the model failed to detect. While not directly involved in the calculations above, TN (True Negatives) indicates the number of normal instances correctly identified, offering a complete perspective on the model's performance across all potential outcomes.

To further refine our evaluation, we calculate the F1 score based on a selected threshold. This threshold determines the sensitivity of the model to anomalies, balancing the trade-off between precision and recall. Our objective is to identify the optimal F1 score (F1-best) that the model can achieve, which involves exploring all possible anomaly thresholds. This exhaustive search ensures that we pinpoint the threshold that maximizes the model's effectiveness in anomaly detection.

Moreover, our evaluation considers temporal data, where anomalies may occur within specific time windows. Identifying and reporting any subset of these anomalous segments is deemed acceptable. To measure performance under this framework, previous work proposed the point-adjust method (Xu et al., 2018). This method considers a ground truth anomaly segment as correctly detected if any observation within it is classified as anomalous. Detection results for data outside the anomaly segments remain unchanged. In this study, we use the point-adjust method to assess the performance of our model.

5.3. Baselines

This study presents the EH-GAM-EGAN model. In order to evaluate its effectiveness, EH-GAM-EGAN has been benchmarked against a range of leading anomaly detection techniques. The baseline methods selected for comparison are as follows:

- **LOF** (Breunig et al., 2000) is an effective method for identifying outliers by evaluating the local density deviation of a given data point relative to its neighbors. This approach assigns an outlier score to each point, indicating its level of isolation from surrounding data points.
- **Isolation Forest** (Liu, Ting, & Zhou, 2008) is an algorithm that identifies anomalies by isolating them from normal data points. It employs a forest of random trees to partition the data, assuming that anomalies are easier to isolate than normal points.
- **OmniAnomaly** (Su et al., 2019) is a stochastic recurrent neural network developed for effective anomaly detection in multivariate time series. This model learns the typical patterns of the time series data and flags deviations as anomalies.

- **MADGAN** (Li et al., 2019) excels in detecting anomalies by considering inter-dimensional relationships, resulting in superior performance on complex, real-world datasets. To ensure consistency with our experimental setup, we modify MADGAN by replacing its RNN layer with a MLP.
- **USAD** (Audibert et al., 2020) is an anomaly detection framework for multivariate time series that leverages adversarial training of an encoder–decoder architecture. This method improves the model's capacity to highlight reconstruction errors associated with anomalous inputs, striking a balance between detection performance and computational efficiency.
- **MTAD-GAT** (Zhao et al., 2020) employs two GAT modules to capture spatial and temporal dependencies in MTS. Additionally, it combines prediction- and reconstruction-based models to predict and reconstruct the data, and computes anomaly scores based on the reconstruction and prediction errors.
- **GDN** (Deng & Hooi, 2021) utilizes graph structure learning and attention mechanisms to effectively identify anomalies in high-dimensional sensor data by detecting deviations from typical behavior patterns.
- **Anomaly Transformer** (Xu et al., 2022) employs a transformer with an anomaly attention mechanism to capture the differential associations between time points in the time series, thereby enabling anomaly detection and achieving high accuracy.
- **Dcdetector** (Yang et al., 2023) is an anomaly detection model that normalizes multivariate time series and employs a dual attention mechanism to capture multiple perspectives. It detects anomalies by measuring representation discrepancies and calculates anomaly scores based on these differences, ensuring accurate detection.
- **ImDiffusion** (Chen et al., 2023) combines time series imputation with diffusion models, enhancing accuracy and robustness by effectively capturing temporal dependencies and leveraging denoised outputs to improve anomaly prediction.
- **WPS** (Qi et al., 2023) is an unsupervised deep learning model for MTSAD. It combines Wasserstein-GAN with gradient penalty and an effective scoring mechanism to address challenges related to insufficient labeling and complex temporal correlations.
- **CARLA** (Darban et al., 2025a) leverages existing generic knowledge regarding diverse types of time series anomalies in the preliminary stage (Pretext). Subsequently, to differentiate between the representations of genuine anomalous windows and those that are normal, a self-supervised classification strategy is employed. This strategy classifies window representations as either normal or anomalous, contingent upon their relative proximity or remoteness from adjacent representations within the feature space during the subsequent phase (self-supervised classification).

This comparative analysis aims to highlight the strengths and potential areas for improvement of the EH-GAM-EGAN model relative to these established methods. By evaluating performance across a range of metrics and datasets, we aim to demonstrate the efficacy of EH-GAM-EGAN in addressing the unique challenges of MTSAD.

5.4. Experimental setup

This section outlines the configuration and parameters employed in the experiments, providing clarity on the computational environment and the architectural specifics of the models under consideration.

- Computational and Model Environment: The experiments were conducted using PyTorch version 1.7.0, selected for its robustness in managing complex computations and its support for GPU acceleration via CUDA 11.0. Model training was efficiently executed on an NVIDIA 3060 GPU, leveraging its parallel processing capabilities. For implementing graph neural networks, a critical component of the graph attention-based temporal prediction module, the PyTorch Geometric Library version 2.1.0 (Fey & Lenssen, 2019) was utilized, providing optimized tools for processing graph-structured data.
- Model Specifics and Training Details: The time series forecasting models employed a historical window size of 6, striking a balance between providing sufficient historical context and maintaining computational efficiency. The graph attention model utilized two layers with dual attention heads to effectively aggregate information from neighboring nodes. Meanwhile, the LSTM-based model was designed with a hidden dimension of 64, enabling it to capture temporal dependencies without introducing excessive complexity. To enhance generalization and mitigate overfitting, regularization techniques such as Dropout (with a rate of 0.2) and Batch Normalization were applied.
- GAN Architecture and Optimization: The architecture for the GAN included a discriminator with a one-dimensional convolutional layer to maintain the original dimensionality of the time series data effectively. Both the generator and discriminator were structured as three-layer MLPs, where each layer was designed to half the dimension of the input time series, striking a balance between complexity and feature extraction capability. The training of these models utilized the Adam optimizer to manage sparse gradients and adapt learning rates effectively. Hyperparameters were carefully selected, with $\lambda = 1$ balancing different loss components and $n_{\text{critic}} = 1$ to maintain training equilibrium. The initial learning rates were set conservatively at $1e{-}4$ for both the generator and discriminator to ensure stable convergence.
- Learning Rates and Anomaly Detection: For the prediction models, learning rates were adjusted according to the dataset complexities, set at $2e{-}3$ for the MSL datasets and $1e{-}4$ for other datasets. During the anomaly detection phase, the parameters of the AD_{score}, including $\alpha = 0.15$ and $\beta = 0.35$, were empirically determined to optimize the model's sensitivity to anomalies, with γ and δ both set at 0.25, further refining the detection capabilities.

5.5. Experimental results

The EH-GAM-EGAN model is introduced in this study as a novel approach designed to address the complexities inherent in detecting anomalies across diverse datasets. The model performance against a suite of advanced anomaly detection techniques is evaluated using precision, recall, and F1 score as the primary metrics for comparison. This analysis is predicated on ten iterations of testing across each dataset, which ensures the reliability and robustness of the performance indicators obtained.

5.5.1. Baseline comparisons

The results in Table 3 demonstrate that the EH-GAM-EGAN model achieves superior performance compared to all other methods, as evidenced by an average F1 score of 0.9413 across the four datasets. This surpasses the performance of the next-best method and underscores the model's efficacy and substantial advancements over existing approaches. Specifically, the model achieves average improvements of 17.93% in precision, 17.88% in recall, and 21.46% in F1 score. Moreover, the robustness of EH-GAM-EGAN is evident from its consistently high precision and recall values, which exceed 0.8 and 0.85, respectively, across all datasets.

In the context of analyzing datasets from cloud platforms such as SMD and PSM, algorithms including LOF and iForest prove insufficient for handling high-dimensional and nonlinear data, attributable to their use of elementary distance metrics and decision procedures. The recall rate for MADGAN remains at an acceptable level. Despite this, the model, which is based on Generative Adversarial Networks, is afflicted by the common issues of vanishing gradients and mode collapse, significantly diminishing its precision and F1 score. OmniAnomaly, identified

Table 3
Comparative performance of the EH-GAM-EGAN model against baseline methods on multiple datasets. The highest performing result for each metric is denoted in **bold**, while the second highest is indicated by an underline.

Method	SMD (subset)			PSM (subset)			SMAP (subset)		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
LOF	0.6678	0.2626	0.3770	0.5789	0.9049	0.7061	0.5893	0.5633	0.5760
iforest	0.7798	0.8860	0.8295	0.1927	0.4793	0.2749	0.5239	0.5907	0.5553
OmniAnomaly	0.8740	0.9716	0.9202	0.7599	0.8496	0.8022	0.9249	0.8199	0.8692
MADGAN	0.6305	0.8959	0.7401	0.8204	0.8034	0.8118	0.8747	0.8049	0.8214
USAD	0.6795	0.9115	0.7786	**1.0000**	0.2091	0.3458	0.2837	**1.0000**	0.4420
MTAD-GAT	0.8828	0.8492	0.8657	0.9528	0.7565	0.8434	0.8906	0.9123	0.9013
GDN	0.2270	0.1510	0.1810	0.9541	0.9050	0.9289	0.8935	0.9220	0.9075
AnomalyTrans	0.8260	0.9355	0.8773	0.9582	0.9713	0.9647	0.9367	0.9913	0.9632
Dcdetector	0.8361	0.9086	0.8708	0.9619	0.9708	0.9663	**0.9443**	0.9841	**0.9638**
ImDiffusion	0.9520	0.9509	0.9488	0.9811	0.9753	**0.9781**	0.8771	0.9618	0.9175
WPS	**0.9920**	0.9355	0.9639	0.9735	0.9001	0.9342	0.6777	**1.0000**	0.8079
CARLA	0.7021	0.8575	0.7721	0.9439	0.8746	0.9080	0.7582	0.9863	0.8573
EH-GAM-EGAN	0.9622	**0.9825**	**0.9723**	0.9466	**0.9845**	0.9651	0.8343	**1.0000**	0.9096

Method	MSL (subset)			Overall		
	Pre	Rec	F1	mean (Pre)	mean (Rec)	mean (F1)
LOF	0.5789	0.9049	0.7061	0.5783	0.6458	0.5677
iforest	0.7928	0.8593	0.8247	0.5723	0.7038	0.6211
OmniAnomaly	0.6974	0.8028	0.7464	0.8141	0.8610	0.8345
MADGAN	0.8078	0.8634	0.8347	0.7834	0.8419	0.8020
USAD	0.7677	0.9282	0.8404	0.6827	0.7622	0.6017
MTAD-GAT	0.8754	0.9440	0.9084	0.9004	0.8655	0.8797
GDN	0.7908	**0.9950**	0.8812	0.7164	0.7433	0.7247
AnomalyTrans	0.9174	0.9473	0.9322	0.9096	0.9614	0.9344
Dcdetector	0.9222	0.9448	**0.9477**	0.9161	0.9521	0.9372
ImDiffusion	0.8930	0.8638	0.8779	**0.9258**	0.9380	0.9306
WPS	**0.9516**	0.8084	0.8742	0.8987	0.9110	0.8950
CARLA	0.8657	0.9583	0.9096	0.8175	0.9192	0.8618
EH-GAM-EGAN	0.8949	0.9429	0.9183	0.9095	**0.9774**	**0.9413**

Table 4
Comparison of training durations for EH-GAM-EGAN and baseline models.

Method	Average training time per epoch (s)
OmniAnomaly	209.78
USAD	168.14
MADGAN	172.46
EH-GAM-EGAN	32.04

as a stochastic recurrent neural network, is designed to encapsulate complex temporal dependencies within time series data. Its less effective performance in specific scenarios may be partly due to the disregard for explicit spatial dependencies among the random variables. AnomalyTrans introduces a novel anomaly detection criterion that incorporates association differences to distinguish anomalous data from normal data. However, it fails to account for spatial dependencies in multivariate time-series data, which limits its anomaly detection performance. In contrast, Dcdetector considers both spatial and temporal dependencies, resulting in a moderate improvement in detection performance. Our method, by integrating discriminative errors and accounting for both spatial and temporal dependencies, achieves superior performance. As shown in Table 3, on the SMD dataset, EH-GAM-EGAN outperforms AnomalyTrans by 10.82% and Dcdetector by 11.65%. In terms of the average F1 score across four datasets, EH-GAM-EGAN exceeds AnomalyTrans by 0.74% and Dcdetector by 0.44%. ImDiffusion integrates time series imputation with diffusion models. As shown in Table 3, while ImDiffusion performs exceptionally well on the PSM dataset, EH-GAM-EGAN outperforms it by 2.48% on the SMD dataset. Additionally, EH-GAM-EGAN exceeds ImDiffusion by 1.15% in the average F1 score across the four datasets. This discrepancy is likely attributed to the smaller size of the PSM dataset, whereas the SMD dataset is larger. ImDiffusion may face performance limitations when processing larger datasets, resulting in inferior detection results compared to EH-GAM-EGAN. It is clear that our model is better suited for handling data from cloud platforms, characterized by complex graph structures and larger data volumes.

In the comparison of graph-based methods, GDN, an approach based on attention-driven graph neural networks, effectively captures explicit spatial dependencies among random variables. However, its performance may be limited in certain situations due to its neglect of explicit temporal dependencies between these variables. MTAD-GAT addresses both spatial and temporal dependencies, computing anomaly scores based on reconstruction and prediction errors. Our method, in contrast, not only computes anomaly scores through reconstruction and prediction errors but also leverages an enhanced GAN framework. This framework, by utilizing the dual mechanisms of generative and discriminative models, strengthens the robustness and accuracy of anomaly detection, resulting in superior performance. As shown in Table 3, EH-GAM-EGAN outperforms GDN by 29.88% and MTAD-GAT by 7.00% in terms of the average F1 score across four datasets.

A comprehensive analysis of the data reveals that the improvements in the F1 score for the SMAP and MSL datasets are less significant compared to those for the SMD and PSM datasets. This discrepancy is attributed to the inherent differences in the nature of the datasets. Specifically, NASA anomaly datasets, such as SMAP and MSL, measure various attributes like radiation, temperature, and computational activity. While these attributes are not completely independent, their interrelationships are weaker than those found in cloud-based datasets like SMD and PSM, where feature dependencies are naturally stronger (Gao et al., 2024; Zhou et al., 2024). The EH-GAM-EGAN model, which combines graph attention mechanisms with enhanced generative adversarial networks, excels in analyzing datasets with closely correlated features, demonstrating its potential for application in cloud-based environments.

An essential aspect of evaluating the practicality of anomaly detection models is their training efficiency. Table 4 provides a comparative overview of the average training time per epoch for each model, thereby shedding light on the computational demands associated with each approach.

It is noteworthy that the EH-GAM-EGAN model exhibits an average training time of 32.04 s per iteration, which represents a significant

Table 5
Results of parameter sensitivity tests for α and β.

α	β	γ	δ	SMD	PSM	SMAP	MSL	Overall
0.05	0.45	0.25	0.25	0.9678	**0.9158**	0.8996	0.8486	0.9080
0.15	0.35	0.25	0.25	**0.9709**	0.9094	**0.9056**	**0.8649**	**0.9127**
0.25	0.25	0.25	0.25	0.9576	0.9100	0.8680	0.8577	0.8983
0.35	0.15	0.25	0.25	0.9529	0.8996	0.8447	0.6693	0.8416
0.45	0.05	0.25	0.25	0.9505	0.9017	0.8151	0.6423	0.8274
0.50	0.00	0.25	0.25	0.9477	0.8945	0.8031	0.6396	0.8212

Table 6
Results of parameter sensitivity tests for γ and δ.

α	β	γ	δ	SMD	PSM	SMAP	MSL	Overall
0.15	0.35	0.05	0.45	0.9571	0.8729	0.6565	0.8637	0.8375
0.15	0.35	0.15	0.35	0.9616	0.8772	0.8938	0.8607	0.8983
0.15	0.35	0.25	0.25	**0.9709**	0.9094	**0.9056**	0.8649	**0.9127**
0.15	0.35	0.35	0.15	0.9512	0.9153	0.8870	0.8286	0.8955
0.15	0.35	0.45	0.05	0.9161	0.9151	0.8938	**0.8883**	0.9033
0.15	0.35	0.50	0.00	0.8959	**0.9157**	0.7976	0.8321	0.8604

improvement in efficiency compared to its counterparts. This efficiency represents an approximate 85% reduction in training time compared to the MADGAN model, which recorded the longest training duration. This significant enhancement in training efficiency not only highlights EH-GAM-EGAN's superior performance but also expands its applicability to large-scale datasets.

The experimental results demonstrate the exceptional capability of the EH-GAM-EGAN model in anomaly detection within multivariate time series data. Its superior performance, coupled with remarkable training efficiency, positions the EH-GAM-EGAN model as a formidable tool in the anomaly detection landscape, offering significant advancements over existing methodologies.

5.5.2. Parameter sensitivity analysis

This section aims to investigate the impact of weighting parameters on the performance of the EH-GAM-EGAN model in MTSAD. The Eq. (18) includes these parameters, which determine the significance of prediction errors associated with α for graph attention-based (PS_{score}), β for LSTM-based (PT_{score}), γ for reconstruction (G_{score}), and δ for discrimination (D_{score}).

In order to assess the sensitivity of the EH-GAM-EGAN model to these parameters, a comprehensive analysis was conducted, in which α and β were varied while γ and δ were kept fixed at 0.25. Furthermore, we evaluated the performance of the EH-GAM-EGAN model under fixed values of α and β, while varying γ and δ. The evaluation encompassed a number of scenarios, with F1 scores calculated for each dataset. In order to derive a universal metric, the F1 scores were averaged across all datasets, resulting in an overall F1 performance metric.

Table 5 showcases the F1 scores for different combinations of α and β. Notably, the combination of $\alpha = 0.15$ and $\beta = 0.35$ yields the highest F1 scores across all datasets, achieving an overall F1 score of 0.9127. This combination outperforms other parameter combinations, demonstrating its efficacy in enhancing the model's anomaly detection capabilities. Similarly, Table 6 explores the impact of varying γ and δ values while keeping α and β fixed at $\alpha = 0.15$ and $\beta = 0.35$. The combination of $\gamma = 0.25$ and $\delta = 0.25$ consistently produces the highest F1 scores across all datasets, resulting in an overall F1 score of 0.9127. This finding further reinforces the importance of these parameters in optimizing the EH-GAM-EGAN model's performance.

Fig. 8 further illustrates the F1 score dynamics in response to variations in α and β across four distinct datasets. The combined F1 score is the mean of these datasets' scores. To elucidate the influence of the parameters, we restricted the sum of α and β to 0.5 and varied each within the 0 to 0.5 interval. The analysis reveals that an excessively low α (e.g., 0.05) leads to the model's underperformance due to inadequate capture of the interdependencies in multivariate time series. Conversely, a high α (e.g., 0.5) also does not yield optimal anomaly

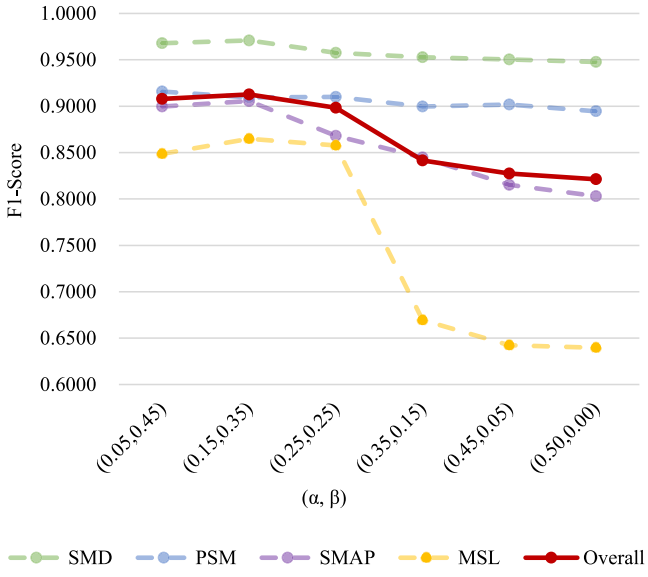

Fig. 8. Parameter sensitivity tests were conducted by adjusting α and β, which were initially set at 0.05 and 0.45, respectively, maintaining their sum at 0.5. The values were altered in increments of 0.1 for α and corresponding decrements for β. The model's performance, measured by the F1 score, peaked at $\alpha = 0.15$ and $\beta = 0.35$.

detection accuracy, as portrayed in EH-GAT-EGAN's declining F1 score. Our results indicate an optimal balance at $\alpha = 0.15$ and $\beta = 0.35$, which achieves the highest F1 score. Subsequent experiments will therefore employ these values for (α, β).

Fig. 9 also delineates the effect of adjusting γ and δ on the model's F1 score across a suite of four datasets. These parameters were varied from their respective minima at 0 to maxima at 0.5, with the constraint that their sum remains constant at 0.5. This normalization ensures a focused examination of their relative impacts. Specifically, the configuration of $\gamma = 0.25$ and $\delta = 0.25$, when combined with previously optimized α and β values, results in the most advantageous F1 outcomes. This evidence has informed our decision to adopt (α, β, γ, δ) = (0.15, 0.35, 0.25, 0.25) for the remainder of our analyses.

5.5.3. Ablation experiments

The results of our ablation experiments highlight the robustness and effectiveness of the EH-GAM-EGAN model in the field of MTSAD. These findings are presented in Table 7 and Fig. 10, which compare the performance of different model configurations across four benchmark datasets. EH-GAM-EGAN consistently achieves superior performance

Table 7
Ablation experiments: Performance metrics.

Method		w/o PS	w/o PT	w/o D	w/o GP	Ours
SMD	Pre	0.9526	0.9920	0.9019	**0.9953**	0.9622
	Rec	0.9742	0.9044	0.9213	0.9360	**0.9825**
	F1	0.9633	0.9462	0.9115	0.9647	**0.9723**
PSM	Pre	0.9669	0.9694	**0.9806**	0.6652	0.9466
	Rec	0.8487	0.8487	0.8487	0.8496	**0.9845**
	F1	0.9039	0.9050	0.9099	0.7462	**0.9651**
SMAP	Pre	0.7953	0.6570	0.7514	0.6688	**0.8343**
	Rec	1.0000	1.0000	1.0000	1.0000	**1.0000**
	F1	0.8860	0.7930	0.8580	0.8015	**0.9096**
MSL	Pre	0.8324	0.4538	0.8683	0.8497	**0.8949**
	Rec	0.8584	**0.9995**	0.9430	0.8930	0.9429
	F1	0.8452	0.6242	0.9041	0.8708	**0.9183**
Overall	Pre	0.8868	0.7680	0.8756	0.7948	**0.9095**
	Rec	0.9203	0.9381	0.9282	0.9196	**0.9775**
	F1	0.8996	0.8171	0.8959	0.8458	**0.9413**

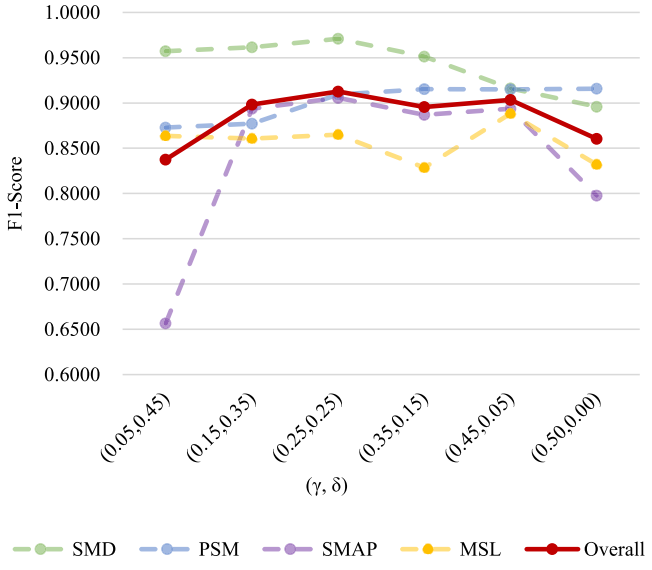

Fig. 9. Parameter sensitivity tests on γ and δ, with fixed values of $\alpha = 0.15$ and $\beta = 0.35$, demonstrated the impact on model performance. Starting at $\gamma = 0.05$ and $\delta = 0.45$, and maintaining their total at 0.5, the parameters were adjusted in increments of 0.1. Optimal model performance, as measured by the F1 score, was achieved at $\gamma = 0.25$ and $\delta = 0.25$.

metrics, with the highest F1 scores, precision, and recall rates across all datasets. This comprehensive performance is highlighted in Fig. 10, confirming the model's overall efficacy. The specifics of model variants analysis are elucidated as follows:

(1) **EH-GAM-EGAN (w/o PS)** : This variant omits the graph attention temporal prediction model, leading to a decline in performance metrics (precision, recall, and F1 score) by an average of 2.50%, 5.85%, and 4.43%, respectively. This reduction demonstrates the critical role of the graph attention mechanism in capturing temporal dependencies for accurate anomaly detection.

(2) **EH-GAM-EGAN (w/o PT)** : By removing the LSTM-based temporal prediction, this model variant shows a more pronounced average performance drop of 15.56% in precision, 4.03% in recall, and 13.19% in F1 score across the datasets. This variant's performance decrement underscores the LSTM's capability in enhancing temporal feature representations, which is crucial for the model's success.

(3) **EH-GAM-EGAN (w/o D)** : The removal of discriminative error components results in a statistically significant decline in performance, with an average drop of 3.73% in precision, 5.04% in recall, and 4.82% in F1 score across the datasets. These figures highlight the crucial role of discriminative error analysis in the scoring phase, which significantly enhances the accuracy of anomaly detection.

(4) **EH-GAM-EGAN (w/o GP)** : the variant employs a classic GAN architecture without gradient penalty, demonstrates a substantial decrease in performance metrics: 12.61% in precision, 5.92% in recall, and 10.15% in F1 score. This finding suggests that the integration of Wasserstein distance and gradient penalty in the GAN architecture is crucial for generating a distribution that closely resembles the real data, thereby facilitating a more accurate anomaly detection process.

These ablation studies highlight the role of individual components to the overall effectiveness of EH-GAM-EGAN, illustrating the synergy between the graph attention mechanism, LSTM temporal predictions, discriminative error handling, and advanced GAN architecture in optimizing anomaly detection tasks.

6. Conclusion

In this study, we present the EH-GAM-EGAN, an unsupervised hybrid model that uniquely integrates a Graph Attention Mechanism (GAM) with an Enhanced Generative Adversarial Network (EGAN). Our experiments show that the EH-GAM-EGAN significantly outperforms existing baseline methods in the domain of MTSAD. By combining GAM and EGAN, our model effectively captures complex data interdependencies and generates high-fidelity time series reconstructions.

When applied to cloud computing infrastructure, the EH-GAM-EGAN efficiently detects anomalies that were previously difficult to identify, overcoming challenges posed by noise and system complexity. The model's anomaly scoring mechanism, which aggregates errors from multiple components, offers a detailed understanding of anomalies, improving the detection of server performance issues with fewer false alarms. Moreover, the inclusion of WGAN-GP ensures stable and efficient training, leading to consistent, high-quality data reconstructions that are essential for identifying potential system failures.

The EH-GAM-EGAN also exhibits reduced training times and improved scalability, with an average iteration time of 32.04 s, making it highly suitable for large-scale cloud computing environments. Extensive testing across diverse datasets, including those from cloud infrastructures and space missions, has confirmed the model's effectiveness. Notably, the EH-GAM-EGAN achieved an average F1 score improvement of 21.46% over baseline models, further reinforcing its reliability for practical applications.

Future research should focus on the following areas: First, improving the model's ability to generate detailed anomaly explanations, potentially through the use of graph structural changes to clarify the underlying causes of anomalies and support decision-making. Second,

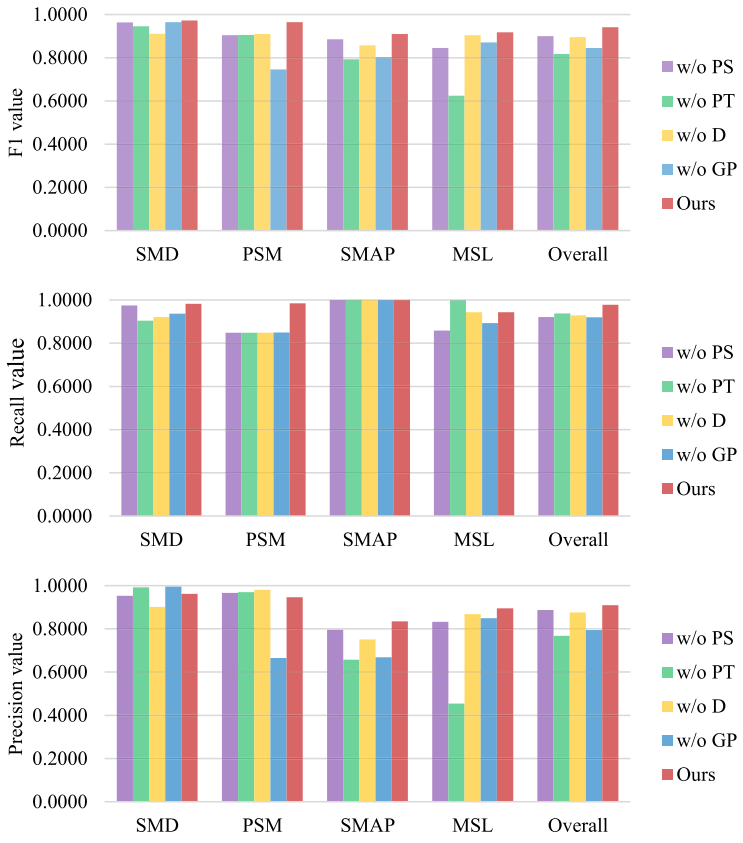

Fig. 10. Ablation experiments: Overall performance visualization.

exploring how the model can be optimized to handle a broader range of anomaly patterns across different scales, particularly in dynamic environments. Finally, future studies should assess the model's cross-domain applicability, emphasizing its scalability and generalizability across various application contexts.

CRediT authorship contribution statement

Yue He: Conceptualization, Data curation, Formal analysis, Writing – original draft. **Xiaoliang Chen:** Funding, acquisition,Investigation, Methodology,Supervision, Writing – review & editing. **Duoqian Miao:** Funding acquisition, Project administration. **Hongyun Zhang:** Validation. **Xiaolin Qin:** Funding acquisition, Resources. **Shangyi Du:** Software, Visualization. **Peng Lu:** Writing – review & editing.

Code availability

Data are available for download at the following web links. https://github.com/huoyoyo/EH-GAM-EGAN

Declaration of competing interest

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

Acknowledgments

This work is supported by the National Key R&D Program of China (Grant nos. 2023YFB3308601,2022YFB3104700), the Science and Technology Program of Sichuan Province (Grant no. 2023YFS0424), the National Natural Science Foundation (Grant nos. 62402395, 623 76198, 62076182), the Science and Technology Service Network Initiative (No. KFJ-STS-QYZD-2021-21-001), and the Talents by Sichuan provincial Party Committee Organization Department, and Chengdu - Chinese Academy of Sciences Science and Technology Cooperation Fund Project (Major Scientific and Technological Innovation Projects). All authors approved the final version of the manuscript.

Data availability

Data will be made available on request.

References

Abdulaal, A., Liu, Z., & Lancewicki, T. (2021). Practical approach to asynchronous multivariate time series anomaly detection and localization. In F. Zhu, B. C. Ooi, & C. Miao (Eds.), *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 2485–2494). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3447548.3467174.

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In T. Elomaa, H. Mannila, & H. Toivonen (Eds.), *European conference on principles of data mining and knowledge discovery* (pp. 15–27). Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/3-540-45681-3_2.

Aslanpour, M. S., Gill, S. S., & Toosi, A. N. (2020). Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. *Internet of Things*, *12*, Article 100273. http://dx.doi.org/10.1016/J.IOT.2020.100273.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2020). Usad: Unsupervised anomaly detection on multivariate time series. In R. Gupta, Y. Liu, J. Tang, & B. A. Prakash (Eds.), *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3395–3404). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3394486.3403392.

Bashar, M. A., & Nayak, R. (2020). Tanogan: Time series anomaly detection with generative adversarial networks. In *2020 IEEE symposium series on computational intelligence* (pp. 1778–1785). IEEE, http://dx.doi.org/10.1109/SSCI47803.2020.9308512.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In W. Chen, J. F. Naughton, & P. A. Bernstein (Eds.), *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 93–104). New York, NY, USA: ACM, http://dx.doi.org/10.1145/342009.335388.

Çavdar, T., Ebrahimpour, N., Kakız, M. T., & Günay, F. B. (2023). Decision-making for the anomalies in iIoTs based on 1D convolutional neural networks and Dempster-Shafer theory (DS-1DCNN). *Journal of Supercomputing*, *79*(2), 1683–1704. http://dx.doi.org/10.1007/S11227-022-04739-2.

Chen, X., Deng, L., Huang, F., Zhang, C., Zhang, Z., Zhao, Y., et al. (2021). Daemon: Unsupervised anomaly detection and interpretation for multivariate time series. In *2021 IEEE 37th international conference on data engineering* (pp. 2225–2230). IEEE, http://dx.doi.org/10.1109/ICDE51399.2021.00228.

Chen, Y., Zhang, C., Ma, M., Liu, Y., Ding, R., Li, B., et al. (2023). ImDiffusion: Imputed diffusion models for multivariate time series anomaly detection. *Proceedings of the VLDB Endowment*, *17*(3), 359–372.

Chen, F., Zhang, Y., Qin, Z., Fan, L., Jiang, R., Liang, Y., et al. (2024). Learning multi-pattern normalities in the frequency domain for efficient time series anomaly detection. In *40th IEEE international conference on data engineering, ICDE 2024, utrecht, the netherlands, May 13-16, 2024* (pp. 747–760). IEEE, http://dx.doi.org/10.1109/ICDE60146.2024.00047.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, October 25-29, 2014, doha, Qatar, a meeting of SIGDAT, a special interest group of the ACL* (pp. 1724–1734). ACL, http://dx.doi.org/10.3115/V1/D14-1179.

Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2025a). CARLA: self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition*, *157*, Article 110874. http://dx.doi.org/10.1016/J.PATCOG.2024.110874.

Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2025b). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, *57*(1), 15:1–15:42. http://dx.doi.org/10.1145/3691338.

Deng, J., Chen, X., Jiang, R., Yin, D., Yang, Y., Song, X., et al. (2024). Disentangling structured components: Towards adaptive, interpretable and scalable time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, *36*(8), 3783–3800. http://dx.doi.org/10.1109/TKDE.2024.3371931.

Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(5), 4027–4035. http://dx.doi.org/10.1609/AAAI.V35I5.16523.

Ergen, T., & Kozat, S. S. (2019). Unsupervised anomaly detection with LSTM neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(8), 3127–3141. http://dx.doi.org/10.1109/TNNLS.2019.2935975.

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch geometric. CoRR arXiv:1903.02428.

Gao, R., He, W., Yan, L., Liu, D., Yu, Y., & Ye, Z. (2024). Hybrid graph transformer networks for multivariate time series anomaly detection. *Journal of Supercomputing*, *80*(1), 642–669. http://dx.doi.org/10.1007/S11227-023-05503-W.

Goodge, A., Hooi, B., Ng, S. K., & Ng, W. S. (2021). Robustness of autoencoders for anomaly detection under adversarial impact. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 1244–1250). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/IJCAI.2020/173.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein GANs. *Advances in Neural Information Processing Systems*, *30*, 5767–5777.

Guo, H., Zhou, Z., Zhao, D., & Gaaloul, W. (2024). EGNN: Energy-efficient anomaly detection for IoT multivariate time series data using graph neural network. *Future Generation Computer Systems*, *151*, 45–56. http://dx.doi.org/10.1016/J.FUTURE.2023.09.028.

Han, S., & Woo, S. S. (2022). Learning sparse latent graph representations for anomaly detection in multivariate time series. In A. Zhang, & H. Rangwala (Eds.), *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2977–2986). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3534678.3539117.

He, Z., Chen, P., Li, X., Wang, Y., Yu, G., Chen, C., et al. (2020). A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(4), 1705–1719. http://dx.doi.org/10.1109/TNNLS.2020.3027736.

Ho, T. K. K., Karami, A., & Armanfard, N. (2023). Graph-based time-series anomaly detection: A survey. CoRR arXiv:2302.00058.

Huang, C., Min, G., Wu, Y., Ying, Y., Pei, K., & Xiang, Z. (2022). Time series anomaly detection for trustworthy services in cloud computing systems. *IEEE Transactions on Big Data*, *8*(1), 60–72. http://dx.doi.org/10.1109/TBDATA.2017.2711039.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Y. Guo, & F. Farooq (Eds.), *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 387–395). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3219819.3219845.

Khanmohammadi, F., & Azmi, R. (2024). Time-series anomaly detection in automated vehicles using D-CNN-LSTM autoencoder. *IEEE Transactions on Intelligent Transportation Systems*, *25*(8), 9296–9307. http://dx.doi.org/10.1109/TITS.2024.3380263.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th international conference on learning representations, ICLR 2017, toulon, France, April 24-26, 2017, conference track proceedings*. OpenReview.net.

Lazarevic, A., & Kumar, V. (2005). Feature bagging for outlier detection. In R. Grossman, R. J. Bayardo, & K. P. Bennett (Eds.), *Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining* (pp. 157–166). New York, NY, USA: ACM, http://dx.doi.org/10.1145/1081870.1081891.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., & Ng, S. (2019). MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *Lecture notes in computer science: vol. 11730, Artificial neural networks and machine learning - ICANN 2019: text and time series - 28th international conference on artificial neural networks, munich, Germany, September 17-19, 2019, proceedings, part IV* (pp. 703–716). Springer, http://dx.doi.org/10.1007/978-3-030-30490-4_56.

Li, G., & Jung, J. J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, *91*, 93–102. http://dx.doi.org/10.1016/J.INFFUS.2022.10.008.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422). IEEE, http://dx.doi.org/10.1109/ICDM.2008.17.

Palmieri, F., Fiore, U., & Castiglione, A. (2014). A distributed approach to network anomaly detection based on independent component analysis. *Concurrency and Computation: Practice and Experience*, *26*(5), 1113–1129. http://dx.doi.org/10.1002/CPE.3061.

Park, D., Hoshi, Y., & Kemp, C. C. (2018). A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters*, *3*(2), 1544–1551. http://dx.doi.org/10.1109/LRA.2018.2801475.

Qi, S., Chen, J., Chen, P., Wen, P., Shan, W., & Xiong, L. (2023). An effective WGAN-based anomaly detection model for IoT multivariate time series. In H. Kashima, T. Idé, & W. Peng (Eds.), *Pacific-Asia conference on knowledge discovery and data mining* (pp. 80–91). Cham: Springer, http://dx.doi.org/10.1007/978-3-031-33374-3_7.

Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., et al. (2019). Time-series anomaly detection service at microsoft. In A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, G. Karypis (Eds.), *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD 2019, anchorage, AK, USA, August 4-8, 2019* (pp. 3009–3017). ACM, http://dx.doi.org/10.1145/3292500.3330680.

Rish, I., et al. (2001). An empirical study of the naive Bayes classifier. *Vol. 3*, In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (pp. 41–46). Seattle, WA, USA;.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471. http://dx.doi.org/10.1162/089976601750264965.

Shanmuganathan, V., & Suresh, A. (2024). Markov enhanced I-LSTM approach for effective anomaly detection for time series sensor data. *International Journal of Intelligent Networks*, *5*, 154–160. http://dx.doi.org/10.1016/J.IJIN.2024.02.007.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., & Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop* (pp. 172–179). IEEE Press.

Song, Y., Xin, R., Chen, P., Zhang, R., Chen, J., & Zhao, Z. (2023). Identifying performance anomalies in fluctuating cloud environments: A robust correlative-GNN-based explainable approach. *Future Generation Computer Systems*, *145*, 77–86. http://dx.doi.org/10.1016/J.FUTURE.2023.03.020.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, & G. Karypis (Eds.), *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2828–2837). ACM, http://dx.doi.org/10.1145/3292500.3330672.

Tang, C., Xu, L., Yang, B., Tang, Y., & Zhao, D. (2023). GRU-based interpretable multivariate time series anomaly detection in industrial control system. *Computers & Security*, *127*, Article 103094. http://dx.doi.org/10.1016/J.COSE.2023.103094.

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, *3*(2), 334–340. http://dx.doi.org/10.1016/j.fcij.2018.10.003.

Wang, Z., Jiang, R., Cai, Z., Fan, Z., Liu, X., Kim, K., et al. (2021). Spatio-temporal-categorical graph neural networks for fine-grained multi-incident co-prediction. In G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, H. Tong (Eds.), *CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, queensland, Australia, November 1 - 5, 2021* (pp. 2060–2069). ACM, http://dx.doi.org/10.1145/3459637.3482482.

Wen, P., Yang, Z., Wu, L., Qi, S., Chen, J., & Chen, P. (2022). A novel convolutional adversarial framework for multivariate time series anomaly detection and explanation in cloud environment. *Applied Sciences*, *12*(20), 10390. http://dx.doi.org/10.3390/app122010390.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., et al. (2018). Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In P. Champin, F. Gandon, M. Lalmas, P. G. Ipeirotis (Eds.), *Proceedings of the 2018 world wide web conference on world wide web, WWW 2018, lyon, France, April 23-27, 2018* (pp. 187–196). ACM, http://dx.doi.org/10.1145/3178876.3185996.

Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly transformer: Time series anomaly detection with association discrepancy. In *The tenth international conference on learning representations, ICLR 2022, virtual event, April 25-29, 2022*. OpenReview.net.

Yang, Y., Zhang, C., Zhou, T., Wen, Q., & Sun, L. (2023). Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In A. K. Singh, Y. Sun, L. Akoglu, D. Gunopulos, X. Yan, R. Kumar, F. Ozcan, & J. Ye (Eds.), *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, KDD 2023, long beach, CA, USA, August 6-10, 2023* (pp. 3033–3045). ACM, http://dx.doi.org/10.1145/3580305.3599295.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In J. Lang (Ed.), *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 3634–3640). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/IJCAI.2018/505.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., et al. (2020). Multivariate time-series anomaly detection via graph attention network. In C. Plant, H. Wang, A. Cuzzocrea, C. Zaniolo, X. Wu (Eds.), *20th IEEE international conference on data mining, ICDM 2020, sorrento, Italy, November 17-20, 2020* (pp. 841–850). IEEE, http://dx.doi.org/10.1109/ICDM50108.2020.00093.

Zhou, X., Dai, C., Wang, W., & Qiu, T. (2024). Global-local association discrepancy for multivariate time series anomaly detection in iIoT. *IEEE Internet Things Journal*, *11*(7), 11287–11297. http://dx.doi.org/10.1109/JIOT.2023.3330696.