Journal of Electronics and Information Science (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2025.100116
ISSN 2371-9524 Vol. 10 Num. 1

DAGAD: Dual Adversarial Learning Graph Anomaly Detection in Multivariate Time Series Data

Huiwen Chen, Mingwei Li*, Yutian Xu, Hongfei Zhang, Siqi Zhang

School of Science, Northeastern University at Qinhuangdao, Qinhuangdao, China
**Corresponding author*

Keywords: Time Series; Anomaly Detection; Adversarial Training; Prediction; Autoencoder

Abstract: The increasing number of high-dimensional time series data poses challenges for traditional anomaly detection methods that rely on supervised approaches. In this paper, we propose a stable and novel method called dual adversarial learning graph anomaly detection, which effectively captures complex data relationships and accurately detects anomalies away from them. Our framework utilizes a graph structure to capture complex relationships between variables, while the dual adversarial training overcomes the inherent limitations of autoencoders. In addition, we incorporate the prediction techniques to enhance the ability to identify anomalies. We evaluate our proposed model on publicly available real-world datasets and compare its performance against various existing methods. The experimental results demonstrate that our method achieves more accurate anomaly detection compared to baseline methods.

1. Introduction

With the rapid growth in the number of devices and sensors in network systems in key areas, protecting them from attacks and ensuring the security of their data have become increasingly important. Many real-world systems involve a large number of interconnected sensors that generate significant amounts of time series data. However, the data may exhibit complex and non-linear correlations, making traditional threshold-based methods inadequate. In such cases, automated anomaly detection methods are necessary to quickly detect anomalies in high-dimensional data and enable operators to diagnose and respond to anomalies promptly [1, 2].

Due to the subjectivity of manual labeling and the high diversity of anomalies, anomaly detection problems mainly focus on unsupervised learning problems. In recent years, numerous machine learning-based methods have been developed, with clustering-based methods like K-Means [3] and DBSCAN [4] being the most common. Other approaches include distance-based methods like K-NN [5], isolation-based methods like Isolation Forest [5] and density-based GMM [6]. However, as the dimension increases, these traditional machine learning methods struggle to capture complex and highly nonlinear relationships in the real world, leading to sub-optimal performance due to the curse of dimensionality.

Recently, there has been significant attention given to the ability of unsupervised anomaly detection methods based on deep learning to infer the correlation between time series and identify

abnormal behaviors [7-10]. One popular approach is the use of reconstruction-based methods such as autoencoder (AE) [11], which is well-suited for anomaly detection. AE maps high-dimensional data into a low-dimensional latent feature space and then reconstructs it back to the original space. Several AE-based variant models have been developed, including Deep Autoencoding Gaussian Mixture Model (DAGMM) [12], Multi-Scale Convolutional Recurrent Encoder Decoder (MSCRED), and UnSupervised Anomaly Detection (USAD) [13]. However, these methods tend to overlook the potential feature space that is highly abstracted from the input data. This will lead to increased uncertainty in these models and a decrease in their ability to accurately identify normal behaviors.

Other deep learning-based methods that have garnered significant interest include those based on generative adversarial networks (GAN) [14] and LSTM-based approaches. However, most of these methods do not explicitly consider the correlation between variables, which may limit their effectiveness in modeling data with multiple potential interrelationships.

Recent studies have proposed prediction-based graph neural networks (GNNs) such as the graph attention network (MTAD-GAT) [15] and the graph deviation network (GDN) [16]. However, most GNNs directly utilize graph structure or compressed graph embedding as input, which results in a loss of data information and limits the representation of highly distinct behaviors among different variables.

To address the challenges, we propose a novel approach called dual adversarial learning graph anomaly detection (DAGAD). This approach focuses on learning the structural relationship between variables. In our framework, the adversarial training structure of the autoencoder ensures that the model amplifies reconstruction errors while maintaining training stability. To demonstrate the efficiency of our approach, we conducted tests using publicly available real-world datasets. The main contributions of our paper can be summarized as follows.

We propose a dual adversarial learning graph anomaly detection approach, which learns the structural graph of dependencies between variables. This method combines the benefits of autoencoder and adversarial training to effectively detect anomalies.

We incorporate the concept of GNN prediction into the latent feature space and simultaneously consider the reconstruction error and prediction error for anomaly scoring, aiming to enhance the recognition capability of the model. Furthermore, we impose a constraint on the latent feature space and derive normal behavior by reducing the uncertainty of the latent feature space.

We perform experiments on public real datasets. The results show that DAGAD detects anomalies more accurately than the baseline method.

The subsequent sections of the paper are organized as follows.

2. Related Work

Time series anomaly detection is a complex and practical task. Traditional methods typically employ various classical techniques such as density-based methods, linear-model-based methods [17], distance-based approaches, isolation-based approaches [18], and regression models [19, 20]. TSI [21] converts the time-series input into graphs and utilizes One-Class SVM [22] to identify anomalies. Finally, classical methods employ variants of Auto-Regressive Integrated Moving Average (ARIMA) [23] to model and detect anomalous behavior. However, these approaches are seldom used for anomaly detection in high-order multivariate time series due to their limited ability to capture volatile time series effectively.

In addition to traditional approaches, recent advancements in unsupervised anomaly detection have seen improvements in deep learning techniques for inferring the correlation between high-dimensional time series data. One such method is the variational autoencoder (VAE) [24], which

uses reconstruction errors as abnormal scores in a probabilistic manner. Another approach is the DAGMM method, which utilizes a deep autoencoding Gaussian mixture model for dimension reduction in the feature space and recurrent networks for temporal modeling. This study employs an autoencoder to compress input data into a latent space, which is then used by a recurrent estimation network to predict the next data point. MSCRED is an RNN-based autoencoder with an attention mechanism based on Convolutional Long Short Term Memory (Convolution-LSTM) [25], using a feature matrix as the object of model reconstruction. LSTM-VAE [26] fuses signals and reconstructs their expected distribution by introducing a progress-based varying prior while modeling the time dependence of time series through the LSTM network. MAD-GAN [27] considers both prediction error and discriminator loss, using the generator to model the distribution of time series. USAD is currently one of the top multivariate time series anomaly detection methods. However, most existing methods do not consider the dependencies between variables, making it challenging to model complex, non-stationary high-dimensional time series data.

To address this limitation, several graph neural networks (GNNs) have been recently proposed for modeling graph-structured data [28]. Generally, GNNs [29] assume that the state of a node is influenced by the states of its neighbors. Graph convolutional networks (GCNs) capture the relationship between each node and its adjacent nodes to model the feature representation of nodes. To handle more complex data, the graph deviation network (GDN) learns the structural information between variable embeddings for prediction and detects anomalies by calculating the deviation from real data. However, most existing methods use the graph structure or compressed graph embedding directly as the model input, which imposes certain limitations.

3. Methodology

We first formalize the problem in Section 3.1. In section 3.2, we give the overall framework of our method. Finally, we describe the implementation process and specific details of the method in section 3.3-3.6.

3.1 Problem Formulation

In this paper, our input data is multivariate time series data, which consists of multiple univariate time series. Each univariate time series records one metric to form a sequence of observed data points. For multivariate time series with N variables, each variable has T time steps, which are denoted by

$$S = \{S_1, S_2, \cdots, S_T\}, S_t = \{s_t^1, s_t^2, \cdots, s_t^N\}, t \in \{1,2,\cdots,T\}$$

Where S_t contains N variables at a given time t.

Following the usual unsupervised anomaly detection formulation, the training data is assumed to consist of only normal data S_{train}. Our goal is to detect anomalies in unknown data S_{test} that have the same variables as S_{train}, but on a separate set of T time steps. Hence, we define the calculation of abnormal scores $A(\hat{x}_t)$ to measure the deviation between S_{test} and S_{train}, and assign labels y_t according to a given threshold, where y_t to denote whether the data at the t-th timestamp of S_{test} is anomalous.

3.2 Overview of the Proposed Framework

The proposed DAGAD method aims to learn the dependencies between variables in the form of graphs, and then detect different degrees of deviation from the normal pattern. Figure 1 provides an overview of our framework. It includes four main parts:

(1) Data preprocessing: normalizes data and divides it into time windows of a specific length L.

(2) Graph structure learning: learns the graph structure that represents the dependencies between variables.

(3) Dual adversarial network training: learns structural pattern of normal data based on dual adversarial networks.

(4) Graph abnormal score: identifies deviations of varying degrees from the normal pattern.

3.3 Data Preprocessing

To make our model more robust, we normalize the time series data (training data and test data) to get it in the range [0,1):

$$S_t \leftarrow \frac{S_t - \min(S)}{\max(S) - \min(S) + \hat{\tau}} \tag{1}$$

Where $\min(S)$ and $\max(S)$ are the minimum and maximum vectors in the time series. $\hat{\tau}$ is a quite small constant to prevent zero-division.

To capture the time correlation, we use a sliding window to obtain the input of the model, that is, a sliding window sample with a length of L at a given time t.

$$W_t = \{S_{t-L+1}, \cdots, S_{t-1}, S_t\}$$

We traverse the original time series to obtain a sliding window sequence $W = \{W_1, \cdots, W_T\}$. Instead of using S directly as training input, we will use W to represent the training input and $\widehat{W}$ to represent the unknown input. For an unknown sliding window input $\widehat{W}_t,\ t > T$, we set a label y_t to indicate whether the window at time t is abnormal. If its anomaly score is greater than a given threshold, it is regarded as an anomaly, $y_t = 1$. Otherwise, it is normal, $y_t = 0$.

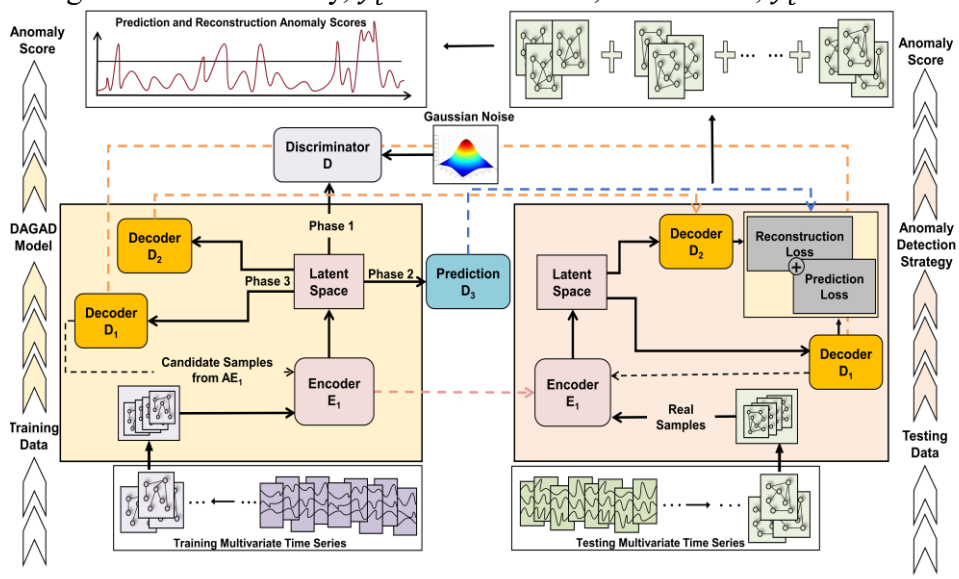

Fig. 1 DAGAD: Dual adversarial learning graph anomaly detection

3.4 Graph Structure Learning

A primary objective of our framework is to learn the relationships between variables in the form of a graph structure. To achieve this, we utilize an undirected graph where each node corresponds to a single variable, and the edges depict hidden dependencies between variables. We use the undirected graph as the model is more concerned about the changes in the relationship between

variables caused by anomalies, and the ease of managing its structure. We use the adjacency matrix A to represent the undirected graph, where A_{ij} represents the existence of undirected edges from node i to node j.

To capture the correlation, we calculate the similarity between node i and other nodes:

$$a_{ij} = a_{ji} = \frac{E\left(w_i^T w_j\right) - E(w_i)E\left(w_j\right)}{\sigma_{wi}\sigma_{wj}} \tag{2}$$

$$A_{ij} = \begin{cases} a_{ij} & a_{ij} > \varepsilon \\ 0 & other \end{cases} \tag{3}$$

Where $i \in (0, N)$, w_i is the corresponding single variable sequence in the window W_t, and ε depicts a given threshold. That is, for each window W_t, we first calculate a_{ij}, the Pearson correlation coefficient between variable i and the remaining variables. Then we select a threshold ε to determine the structural relationship in A: the ε is dynamically adjusted as the model learns, and the user can choose the initial threshold based on the expected sparsity level. In addition, we use the Pearson correlation coefficient to represent node similarity, as it effectively retains the continuously changing trend of data as much as possible, whether the anomaly is significant or subtle. Figure 2 depicts the dynamic learning process of graph structure. Our dual adversarial network is then defined, making use of the learned adjacency matrix A.

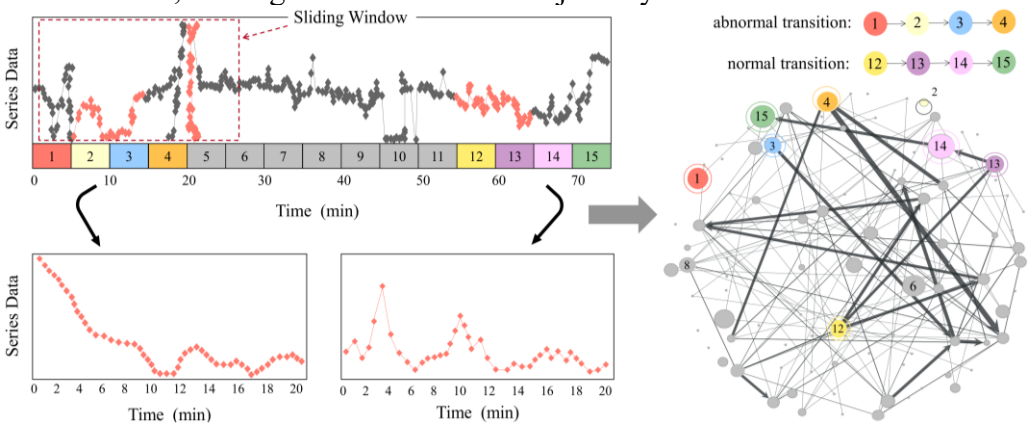

Fig. 2 Time series windowing and graph structure learning process

3.5 Dual Adversarial Network Training

Autoencoder (AE) is an unsupervised neural network. It uses the encoder E to encode the original high-dimensional data X into the potential low-dimensional feature space z and then decodes the high-dimensional results through the decoder D.In anomaly detection, AE performs well in reconstructing normal data, but it yields a higher anomaly score for unknown abnormal samples. However, there are two major challenges in anomaly detection using autoencoders.

The first challenge revolves around the latent feature space. The autoencoder constructs the latent feature space randomly without any boundaries, which causes uncertainty in the reconstruction process. As a result, the performance of the model may be reduced. The second challenge involves detecting subtle anomalies [30]. Since the AE aims to learn the normal data pattern as accurately as possible, it may fail to detect anomalies that are relatively close to the normal data pattern and are therefore small in magnitude.

To address these limitations, we construct a dual adversarial learning framework. On the one hand, this can overcome the inherent limitations of the AE model, while incorporating a prediction component to enhance the anomaly recognition ability. On the other hand, the architecture of the

autoencoder ensures the stability of the adversarial learning process. In addition, we consider the potential representation of the normal graph structure as the embedding of normal patterns. This representation allows us to predict future data structures to some extent, whereas abnormal data cannot be accurately predicted. Thus, we utilize latent features to predict future data structures.
Our framework consists of five parts: an encoder E_1, three decoders D_1, D_2 and D_3, a discriminator D. E_1, D_1, and D_2 constitute the autoencoders AE_1 and AE_2 with the same structure:

$$AE_1(x) = D_1(z), AE_2(x) = D_2(z), z = E_1(x) \tag{4}$$

Where, for the sake of simplicity and without loss of generality, we utilize x to represent the learned adjacency matrix A. E_1 and D are trained in the form of an Adversarial Network, And Here E_1 Is Regarded As A Generator. In Addition, We Use The Decoder Structure D_3 For Prediction, Without Defining A New Network Structure. Further Details Are Provided In The Following.

First Adversarial Component Training. In The First Training Stage, To Reduce The Uncertainty Of The Latent Feature Space, We Aim To Impose A Prior Distribution On The Latent Space (Graph Structure Embedding), That Is Matched With A Normal Distribution $\mathrm{N}(0,1)$. Thus, We Use Adversarial Learning To Provide A Constraint On The Probability Distribution Of Latent Feature Distribution. The Objective Of The Stage Is To Train Discriminator D To Distinguish Between The Embedding And Prior Distribution Generated By E_1, Where E_1 Tries To Minimize $\mathcal{L}_{\mathrm{Adv}}$ Against An Adversary D That Tries To Maximize It. The Objective Is Defined As Follows:

$$\mathcal{L}_{\mathrm{Adv}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}) = \mathrm{E}\left[\mathrm{Log}\left(\mathrm{D}\left(\mathrm{Z}^{'}\right)\right)\right] + \mathrm{E}\left[\mathrm{Log}\left(1 - \mathrm{D}(\mathrm{Z})\right)\right] \tag{5}$$

Where Z Denotes A Normal Graph Structure Embedding, The Shape Of $\mathrm{Z}^{'}$ Is The Same As That Of Z, And Its Elements Are Random Variables Generated According To The Normal Distribution $\mathrm{N}(0,1)$.

Prediction Component Training. In The Second Training Stage, The Training Objective Is To Utilize Latent Features To Predict Future Data Structures. We Take The Graph Embedding From E_1 As Input And Train D_3 To Learn The Normal Graph Structure. The Training Objective Is:

$$\mathcal{L}_{\mathrm{Pre}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_3) = \left\| \mathrm{X}_{\mathrm{T+L}} - \mathrm{X}^{'}_{\mathrm{T+L}} \right\|_2, \quad \mathrm{X}^{'}_{\mathrm{T+L}} = \mathrm{D}_3(\mathrm{Z}_{\mathrm{T}}) \tag{6}$$

Where $\mathrm{X}^{'}_{\mathrm{T+L}}$ Is The Prediction With L Steps After The Graph Structure At Time T (L Represents The Window Size). $\mathrm{X}_{\mathrm{T+L}}$ Is The Real Graph Structure At Time $\mathrm{T} + \mathrm{L}$. Thus, The $\mathrm{X}^{'}_{\mathrm{T+L}}$ Should Be As Close To $\mathrm{X}_{\mathrm{T+L}}$ As Possible.

Second Adversarial Component Training. In The Third Training Stage, This Component Is Designed To Address Another Inherent Limitation Of The Autoencoder, Where AE_1 And AE_2 Have A Dual Purpose. Initially, AE_1 And AE_2 Aim To Reconstruct The Input X Independently, Thus According To Eq. (4), The Training Objectives Are:

$$\mathcal{L}_{\mathrm{Re1}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_1) = \|\mathrm{X} - \mathrm{AE}_1(\mathrm{X})\|_2 \tag{7}$$

$$\mathcal{L}_{\mathrm{Re2}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_2) = \|\mathrm{X} - \mathrm{AE}_2(\mathrm{X})\|_2 \tag{8}$$

Next, We Introduce The Adversarial Training Of AE_1 And AE_2. Here, The Objective Of AE_2 Is To Distinguish Between The Original Input X And The Candidate Reconstruction Generated By AE_1 Through Maximizing The Error. As AE_2, AE_2 Aims To Reconstruct The Original Input $(\mathrm{AE}_2(\mathrm{X}) = \mathrm{X})$ As Much As Possible To Deceive AE_2. This Pushes The AE_2, In This Phase, To Generate The Same Output As $\mathrm{AE}_2(\mathrm{X})$ That Aims To Match The Input X In The Reconstruction Stage.

Hence, The Objective Of AE_1 Is To Minimize The Error Between X And The Reconstructed Output Of AE_2, Whereas AE_2 Aims To Maximize The Same. This Means The Training Objective Is:

$$\mathcal{L}_{\mathrm{Adv1}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_1) = +\left\|\mathrm{X} - \mathrm{AE}_2\left(\mathrm{AE}_1(\mathrm{X})\right)\right\|_2 \tag{9}$$

$$\mathcal{L}_{\mathrm{Adv2}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_2) = -\left\|\mathrm{X} - \mathrm{AE}_2\left(\mathrm{AE}_1(\mathrm{X})\right)\right\|_2 \tag{10}$$

Finally, We Jointly Consider Two Stages To Obtain Evolutionary Loss Functions, Where The Training Objective Of Each AE Is A Combination Of Eq. (7), Eq. (8), Eq. (9) And Eq. (10). The Total Training Loss Is:

$$\mathcal{L}_{\mathrm{AE1}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_1, \mathrm{D}_2) = \mathrm{T}^{-\mathrm{N}}\|\mathrm{X} - \mathrm{AE}_1(\mathrm{X})\|_2 + (1 - \mathrm{T}^{-\mathrm{N}})\left\|\mathrm{X} - \mathrm{AE}_2\left(\mathrm{AE}_1(\mathrm{X})\right)\right\|_2 \tag{11}$$

$$\mathcal{L}_{\mathrm{AE2}}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}_1, \mathrm{D}_2) = \mathrm{T}^{-\mathrm{N}}\|\mathrm{X} - \mathrm{AE}_2(\mathrm{X})\|_2 - (1 - \mathrm{T}^{-\mathrm{N}})\left\|\mathrm{X} - \mathrm{AE}_2\left(\mathrm{AE}_1(\mathrm{X})\right)\right\|_2 \tag{12}$$

Where N Indicates The Training Epoch, Representing The Proportion Of Each Part Evolving, And T Is A Parameter Close To One. It Is Crucial To Remark That In The Initial Part Of The Process, The Reconstruction Loss Is Assigned A Low Weight To Avoid Destabilizing Model Training When The Reconstruction Error Is Large. As The Reconstructions Become Closer To The Inputs, The Weight Of The Adversarial Loss Is Gradually Increased.

Three-Stage Training. In Our Framework, We Combine All The Losses To Obtain The Total Objective Function:

$$\mathcal{L}(\mathrm{X}, \mathrm{E}_1, \mathrm{D}, \mathrm{D}_1, \mathrm{D}_2, \mathrm{D}_3) = \Lambda\mathcal{L}_{\mathrm{Adv}} + \Lambda^{'}\ \mathcal{L}_{\mathrm{Pre}} + \mathcal{L}_{\mathrm{AE1}} + \mathcal{L}_{\mathrm{AE2}} \tag{13}$$

Where Λ And $\Lambda^{'}$ Are Hyperparameters For Balancing The Losses, And $0 < \Lambda, \Lambda^{'} < 1$. The Dual Adversarial Network Is Obtained By Minimizing Eq. (13). We Train The Proposed Framework By Using Stochastic Gradient Descent And Doing Alternative Updates Of Each Component As Follows:

Maximize $\mathcal{L}_{adv}$ by updating weights of D;

Minimize $\mathcal{L}_{adv}$ by updating weights of E_1;

Minimize $\mathcal{L}_{pre}$ by updating weights of D_3;

Minimize $\mathcal{L}_{AE1}$ by updating weights of E_1 and D_1;

Maximize the adversarial loss and minimize the reconstruction loss in $\mathcal{L}_{AE2}$ by updating the weights of E_1 and D_2.

Note that the discriminator D is only utilized during the training phase.

3.6 Graph Abnormal Scoring

Given the learned structure relationships, we want to detect anomalies that deviate from these relationships. To do this, our model combines reconstruction error and prediction error to define the abnormal score:

$$A(\hat{x}_t) = \alpha\|\hat{x}_t - AE_1(\hat{x}_t)\|_2 + \beta\left\|\hat{x}_t - AE_2\left(AE_1(\hat{x}_t)\right)\right\|_2 + \gamma\|\hat{x}_{t+1} - \hat{x}'_{t+1}\|_2 \tag{14}$$

Where $\alpha + \beta = 1$, α, β are hyperparameters used to weigh abnormal sensitivity, and $0 < \gamma < 1$ is used to parameterize the trade-off between reconstruction error and prediction error. We represent $\alpha < \beta$ as a high-detection-sensitivity scenario and $\alpha > \beta$ as a low-detection-sensitivity scenario.

When we obtain the anomaly score for an unknown window, we label this window as anomalous (i.e., $y_t = 1$) if the score exceeds a given threshold. While different approaches could be employed to set the threshold such as peak over threshold (POT) [31] and annual maximum (AM) [32], to

avoid introducing additional hyperparameters, we use in our experiments a simple approach of setting the threshold as the maximum of $A(\hat{x}_t)$ over the validation data.

4. Experiments

In this section, we present the set of experiments conducted to demonstrate the effectiveness of our method. The performance results are compared with state-of-the-art techniques.

4.1 Datasets

We utilize four publicly available datasets in our experiments. It is important to highlight that, despite concerns raised by previous work [33] regarding the suitability of benchmark datasets for multivariate time series anomaly detection, we still use these datasets to facilitate direct comparison with baseline methods. Table 1 summarizes the datasets characteristics and they are briefly described in the following.

Table 1 Benchmarked Datasets. (%) is the percentage of anomalous data points in the dataset

Dataset	Train	Test	Dimensions	Anomalies(%)
SMD	708405	708420	28*38	4.16
SWaT	496800	449919	51	11.98
WADI	1048571	172801	123	5.99

4.2 Evaluation Metrics

We use precision (P), recall (R), and F1-Score (F1) to evaluate the performance of anomaly detection for all models:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

Where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives. If a model lacks a predefined selection threshold, our method (maximum value of the validation set) is utilized for selection. In addition, we hope to compute a model's best F1 score, we calculate all possible anomaly thresholds to search for the best F1, denoted as $\mathrm{F1}_{best}$.

In real-world scenarios, anomalies usually occur continuously, resulting in a stream of abnormal observations. Therefore, it is acceptable to trigger an anomaly alert within any subset of an anomaly segment. Hence, [7] proposed a point adjustment method to calculate performance. As this approach describes, whenever at least an anomaly is detected in an abnormal segment, all points in this segment are considered as anomalies, even if they are not. In this paper, we adopt the point adjustment method to calculate the performance index, which provides a more realistic evaluation standard.

Table 2 Performance comparison of DAGAD with baseline methods on the complete dataset

Methods		SWaT			WADI			SMD	
	P	R	F1	P	R	F1	P	R	F1
AE	0.9913	0.7040	0.8233	0.3970	0.3220	0.3556	0.8825	0.8037	0.8280
L-VAE	0.7123	0.9258	0.8051	0.4632	0.3220	0.3799	0.8698	0.7879	0.8083
DAGMM	0.8292	0.7674	0.7971	0.2228	0.1976	0.2049	0.6730	0.8450	0.7231
MSCRED	0.9992	0.6770	0.8072	0.2513	0.7319	0.3741	0.7276	0.9974	0.8414
MAD-GAN	0.9697	0.6957	0.8101	0.2871	0.5804	0.3842	0.9991	0.8440	0.9150
USAD	0.9870	0.7402	0.8460	0.6451	0.3220	0.4296	0.9314	0.9617	0.9382
DAGAD	0.9882	0.7559	0.8588	0.6630	0.3588	0.4526	0.9871	0.8501	0.9327

5. Results and Analyses

In this section, we present the performance and effectiveness of our proposed DAGAD model in three aspects: overall performance, ablation study, and parameter sensitivity.

5.1 Overall Performance

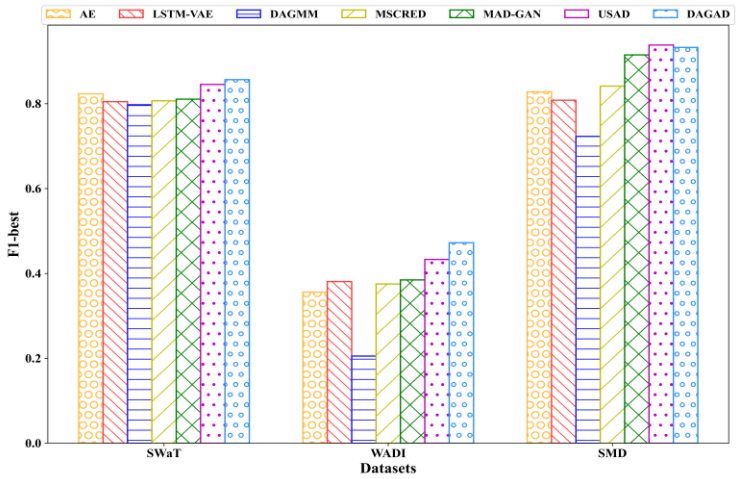

Fig. 3 $F1_{best}$ of DAGAD and all baseline models

To illustrate the overall performance of DAGAD, we compare it with six popular multivariate time series anomaly detection methods, namely AE, LSTM-VAE, DAGMM, MSCRED, MAD-GAN, and USAD. Each of these approaches provides a specific method for selecting anomaly thresholds, and F1 is calculated accordingly. Table 2 provides a detailed overview of the evaluation results for all methods on three public datasets. The results show that DAGAD outperforms all baseline models. It is important to note that all models perform relatively poorly on WADI due to its large scale in terms of sequence lengths and data modality. However, the overall performance improvement of DAGAD over baseline methods is significant, largely as, when dealing with high-dimensional (123-dimensional) data features, the graph structure is capable of deeply extracting the complex relationships between features of normal data, effectively achieving 'data augmentation'. The F1 score of DAGAD on the SMD dataset is slightly lower than the optimal baseline due to the smaller number of features and their similarity.

In addition, Figure 3 illustrates the $F1_{best}$ corresponding to the optimal threshold. All models exhibit no significant difference between F1 and $F1_{best}$ on the SWaT and SMD datasets. However, for the WADI dataset, all methods have improved performance, especially DAGAD is 0.0171 higher than F1 on $F1_{best}$ (0.4725 vs 0.4526). This may be due to some unknown abnormal segments in WADI training data (normal data).

Overall, DAGMM performs poorly on three datasets as it does not use sequence windows but only a single GRU model, regardless of time characteristics. For time series data, observations are collected over time, resulting in a significant time dependence between the data. Methods such as AE, LSTM-VAE, MAD-GAN, MSCRED, and USAD all utilize sequential observations as input, allowing them to preserve time information. Such methods perform reconstruction regardless of anomalous data, which prevents them from detecting subtle anomalies close to the normal trends. In contrast, DAGAD employs a dual adversarial structure training approach to amplify subtle anomalies. Consequently, even in datasets like SMD, DAGAD can detect subtle anomalies in the case of a small gap.

5.2 Ablation Analysis

To study the necessity of each component in our model, we exclude each major component and observe how the model performance changes in each dataset. First, we consider a dual adversarial learning architecture without graph structure, which directly takes original time-series windows as the model input. Second, we consider the model without the predictive assistance module, i.e., we fix $\lambda' = 0$. Third, we study the model without potential spatial constraint, i.e., we fix $\lambda = 0$. Finally, we consider the model without the second adversarial loss, i.e., a single-phase inference and only the reconstruction loss and the prediction loss for model training. All experimental results are summarized in Table 3 and provide the following findings:

When we remove the graph structure, the average drop in F1 scores is 6.3%. This drop is more pronounced for the WADI dataset (15.4%), indicating that for high-dimensional datasets, a graph structure capable of mining variable relationships is necessary.

Removing the predictive assistance module results in an average F1-score decrease of 3.9%, which suggests that the predictive module contributes to performance improvement.

Not having the potential spatial constraint in the model has little effect on the F1 scores ($\approx$ 2.1%).

Removing the second adversarial training primarily affects the SMD and WADI datasets, as a significant portion of the anomalies in these datasets are subtle, and the adversarial loss helps to amplify the anomaly scores. In this scenario, the average decrease in the F1 score is 5.7%.

Table 3 Ablation Study - F1 scores for DAGAD and its ablated versions

Methods	SWaT	WADI	SMD
DAGAD	0.8588	0.4526	0.9327
w/o graph structure	0.8418	0.2991	0.9144
w/o predict module	0.8453	0.4076	0.9195
w/o first adversarial module	0.8517	0.4378	0.9203
w/o second adversarial module	0.8371	0.3956	0.8562

5.3 Sensitivity Analysis

In this section, we study the effect of six parameters, the window size L, λ and λ' of training objective, and α, β and γ of anomaly score. All experiments were done using the SWaT dataset.

Sensitivity to the window size. Figure 4(a) illustrates the performance of the DAGAD model and its variants with different window sizes. The results show that the window size has a significant influence on the overall performance of the model. Smaller windows enable DAGAD to detect anomalies more quickly. However, if the window is too small, it fails to effectively utilize local contextual information. Conversely, with too large a window, short anomalous segments may be obscured by the sheer number of data points within the window. Moreover, larger windows may result in predicted graph structures that contain redundant future information, which can negatively impact the performance of our model.

Sensitivity to the training objective parameter. We also demonstrate the performance of DAGAD and its variants with varying values λ, while λ' remains fixed in Figure 4(b). Similarly, Figure 4(c) shows the performance of DAGAD and its variants at different values λ', while λ is fixed. It is evident that the choice of optimal parameters is varying for different datasets. Notably, increasing both λ and λ' leads to an overall improvement in performance. However, excessively large proportions of λ and λ' can result in longer training stabilization time or affect the parameter update of the primary inference module (Eq. 14). Hence, we set $\lambda = \lambda' = 0.4$ for SWaT, $\lambda =$

0.5, $\lambda' = 0.4$ for WADI, and $\lambda = \lambda' = 0.2$ for SMD.

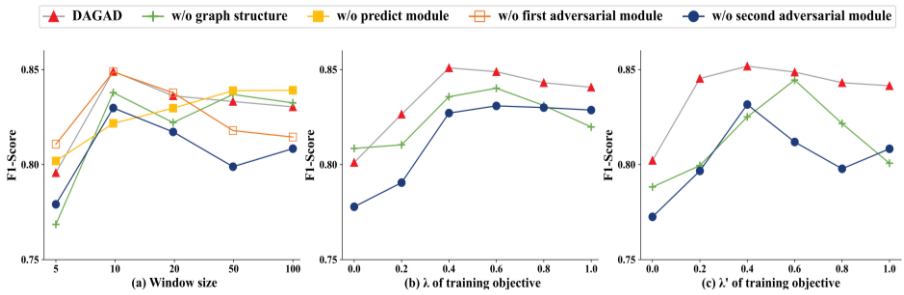

Fig. 4 Effect of parameters

F1-score as a function of A) the window size L, B) λ of the training objective and C) λ' of the training objective in the training set

Sensitivity to the abnormal score parameter. Table 4 reports the effect of different α, β, and γ on F1 score. We observed that by adjusting the values of α and β, the model's sensitivity to anomalies can be fine-tuned. In addition, there are significant potential spatial differences in different data, and we adjust γ to assist the model in achieving the best performance. This ensures that our model meet various requirements for different tasks, making it highly valuable for practical applications.

Table 4 Anomaly detection results with various abnormal score thresholds for SWaT dataset

α	β	γ	F1
0.1	0.9	0.2	0.8486
0.1	0.9	0.4	0.8512
0.4	0.6	0.4	0.8462
0.4	0.6	0.8	0.8423
0.7	0.3	0.8	0.8369

6. Conclusions

We propose a new DAGAD model, which can detect anomalies in multivariate time series data. The learned graph structure is capable of uncovering complex relationships between variables, enabling the model to achieve excellent performance on high-dimensional datasets. DAGAD addresses the limitations of autoencoders through dual adversarial training, while ensuring training stability. In addition, the model incorporates a predictive auxiliary component to enhance its detection capabilities. Finally, experiments show that the DAGAD model has good performance.

References

[1] X. Tang, S. Zeng, F. Yu, W. Yu, Z. Sheng, Z.J.N. Kang, Self-supervised anomaly pattern detection for large scale industrial data, Neurocomputing, 515 (2023) 1-12.

[2] N. Twomey, H. Chen, T. Diethe, P.J.N. Flach, An application of hierarchical Gaussian processes to the detection of anomalies in star light curves, Neurocomputing, 342 (2019) 152-163.

[3] S. Chawla, A. Gionis, k-means–: A unified approach to clustering and outlier detection, Proceedings of the 2013 SIAM international conference on data mining, (SIAM2013), pp. 189-197.

[4] K. Sheridan, T.G. Puranik, E. Mangortey, O.J. Pinon-Fischer, M. Kirby, D.N. Mavris, An application of dbscan clustering for flight anomaly detection during the approach phase, AIAA Scitech 2020 Forum2020), pp. 1851.

[5] Z.K. Maseer, R. Yusof, N. Bahaman, S.A. Mostafa, C.F.M.J.I.a. Foozy, Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset, IEEE access, 9 (2021) 22351-22370. https://doi.org/10.1109/ACCESS.2021.3056614

[6] X. Song, M. Wu, C. Jermaine, S.J.I.T.o.k. Ranka, D. Engineering, Conditional anomaly detection, IEEE

Transactions on knowledge Data Engineering, 19 (2007) 631-645.
[7] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining2019), pp. 2828-2837. https://doi.org/10.1145/3292500.3330672
[8] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G.J.a.p.a. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, ICML 2016 Anomaly Detection Workshop, New York, NY, USA, 20162016). https://doi.org/10.48550/arXiv.1607.00148
[9] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N.V. Chawla, A deep neural network for unsup-evised anomaly detection and diagnosis in multivariate time series data, Proceedings of the AAAI conference on artificial intelligence2019), pp. 1409-1416. https://doi.org/10.48550/arXiv.1811.08055
[10] S. Abilasha, S. Bhadra, P. Deepak, A.J.N. Mathew, Warping resilient scalable anomaly detection in time series, Neurocomputing, 511 (2022) 22-33.
[11] Rumellhart, D. E. "Learning internal representations by error propagation." Parallel distributed processing: explorations in the microstructure of cognition 1 (1986): 319-362.
[12] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, International conference on learning representations2018).
[13] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M.A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining2020), pp. 3395-3404.
[14] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, V. Chandrasekhar, Adversarially learned anomaly detection, 2018 IEEE International conference on data mining (ICDM), (IEEE2018), pp. 727-736. https://doi.org/10.1109/ICDM.2018.00088
[15] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, Multivariate time-series anomaly detection via graph attention network, 2020 IEEE International Conference on Data Mining (ICDM), (IEEE2020), pp. 841-850.
[16] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, Proceedings of the AAAI conference on artificial intelligence2021), pp. 4027-4035.
[17] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, Proce-edings of the IEEE foundations and new directions of data mining workshop, (IEEE Press2003), pp. 172-179.
[18] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, 2008 eighth ieee international conference on data mining, (IEEE2008), pp. 413-422. https://doi.org/10.1109/ICDM.2008.17
[19] P. Boniol, J. Paparrizos, T. Palpanas, M.J.J.P.o.t.V.E. Franklin, SAND: streaming subsequence anomaly detection, Proceedings of the VLDB Endowment, 14 (2021) 1717-1729. https://doi.org/10.14778/3467861.3467863
[20] Y. Wang, N. Masoud, A.J.I.t.o.i.t.s. Khojandi, Real-time sensor anomaly detection and recovery in connected automated vehicle sensors, IEEE transactions on intelligent transportation systems, 22 (2020) 1411-1421. https://doi.org/10.1109/TITS.2020.2970295
[21] M. Fahim, K. Fraz, A.J.I.S. Sillitti, TSI: Time series to imaging based model for detecting anomalous energy consumption in smart buildings, Information Sciences, 523 (2020) 1-13.
[22] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, B.J.I.J.o.E.-H. Furht, M. Communications, Anomaly detection in medical wireless sensor networks using SVM and linear regression models, International Journal of E-Health Medical Communications, 5 (2014) 20-45. https://doi.org/10.1016/j.procs.2015.10.026
[23] A.H. Yaacob, I.K. Tan, S.F. Chien, H.K. Tan, Arima based network anomaly detection, 2010 Second International Conference on Comm-unication Software and Networks, (IEEE2010), pp. 205-209. https://doi.org/10.1109/ICCSN.2010.55
[24] Z. Niu, K. Yu, X.J.S. Wu, LSTM-based VAE-GAN for time-series anomaly detection, Sensors, 20 (2020) 3738. https://doi.org/10.3390/s20133738
[25] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c.J.A.i.n.i.p.s. Woo, Convolutional LSTM network: A machine learning appro-ach for precipitation nowcasting, Advances in neural information processing systems, 28 (2015).
[26] D. Park, Y. Hoshi, C.C.J.I.R. Kemp, A. Letters, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robotics Automation Letters, 3 (2018) 1544-1551. https://doi.org/10.1109/LRA.2018.2801475
[27] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, MAD-GAN: Multivariate anomaly detection for time series data with generative adversa-rial networks, International conference on artificial neural networks, (Springer2019), pp. 703-716.
[28] M. Defferrard, X. Bresson, P.J.A.i.n.i.p.s. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Advances in neural information processing systems, 29 (2016).

https://doi.org/10.48550/arXiv.1606.09375
[29] T.N. Kipf, M.J.a.p.a. Welling, Semi-supervised classification with graph convolutional networks, ICLR 20172016). https://doi.org/10.48550/arXiv.1609.02907
[30] Y. Li, H.S. Abdel-Khalik, A. Al Rashdan, J.J.A.o.N.E. Farber, Feature extraction for subtle anomaly detection using semi-supervised learning, Annals of Nuclear Energy, 181 (2023) 109503. https://doi.org/10.1016/j.anucene.2022.109503
[31] A. Siffer, P.-A. Fouque, A. Termier, C. Largouet, Anomaly detection in streams with extreme value theory, Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining2017), pp. 1067-1075.
[32] N. Bezak, M. Brilly, M.J.H.S.J. Šraj, Comparison between the peaks-over-threshold method and the annual maximum method for flood frequency analysis, Hydrological Sciences Journal, 59 (2014) 959-977. https://doi.org/10.1080/02626667.2013.831174
[33] R. Wu, E.J.I.T.o.K. Keogh, D. Engineering, Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress, IEEE Transactions on Knowledge Data Engineering, (2021). https://doi.org/10.48550/arXiv.2009.13807