Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.048563

ARTICLE

A Novel Graph Structure Learning Based Semi-Supervised Framework for Anomaly Identification in Fluctuating IoT Environment

Weijian Song[1,#], Xi Li[1,#], Peng Chen[1,*], Juan Chen[1], Jianhua Ren[2] and Yunni Xia[3,*]

[1]School of Computer and Software Engineering, Xihua University, Chengdu, 610039, China

[2]West China Second University Hospital, Sichuan University, Chengdu, 610065, China

[3]School of Computer Science, Chongqing University, Chongqing, 400044, China

*Corresponding Authors: Peng Chen. Email: chenpeng@mail.xhu.edu.cn; Yunni Xia. Email: xiayunni@hotmail.com

[#]These authors contributed equally to this work and should be regarded as co-first authors

Received: 12 December 2023 Accepted: 11 April 2024 Published: 08 July 2024

ABSTRACT

With the rapid development of Internet of Things (IoT) technology, IoT systems have been widely applied in healthcare, transportation, home, and other fields. However, with the continuous expansion of the scale and increasing complexity of IoT systems, the stability and security issues of IoT systems have become increasingly prominent. Thus, it is crucial to detect anomalies in the collected IoT time series from various sensors. Recently, deep learning models have been leveraged for IoT anomaly detection. However, owing to the challenges associated with data labeling, most IoT anomaly detection methods resort to unsupervised learning techniques. Nevertheless, the absence of accurate abnormal information in unsupervised learning methods limits their performance. To address these problems, we propose AS-GCN-MTM, an adaptive structural Graph Convolutional Networks (GCN)-based framework using a mean-teacher mechanism (AS-GCN-MTM) for anomaly identification. It performs better than unsupervised methods using only a small amount of labeled data. Mean Teachers is an effective semi-supervised learning method that utilizes unlabeled data for training to improve the generalization ability and performance of the model. However, the dependencies between data are often unknown in time series data. To solve this problem, we designed a graph structure adaptive learning layer based on neural networks, which can automatically learn the graph structure from time series data. It not only better captures the relationships between nodes but also enhances the model's performance by augmenting key data. Experiments have demonstrated that our method improves the baseline model with the highest F1 value by 10.4%, 36.1%, and 5.6%, respectively, on three real datasets with a 10% data labeling rate.

KEYWORDS

IoT multivariate time series; anomaly detection; graph learning; semi-supervised; mean teachers

1 Introduction

With the rapid development and broad application of IoT technology, IoT systems have become an essential part of modern society [1]. In industrial manufacturing, smart homes, medical health, transportation, and logistics, IoT technology plays an important role, bringing convenience to people's

lives and work [2]. However, as the scale of IoT systems continues to expand, their complexity and uncertainty also increase, and abnormal data and security risks gradually emerge [3]. Using deep learning(DL) and machine learning(ML) algorithms [4] for IoT anomaly detection has become a hot research topic in IoT anomaly detection, such as real-time monitoring and rapid response, privacy protection and security, and whether it can effectively detect malicious interference, such as adversarial attacks [5]. Therefore, how designing and implementing effective anomaly detection models to monitor the operation status in IoT systems and respond quickly to abnormal events have become an important research direction [6]. The importance of IoT anomaly detection is self-evident. In IoT systems, abnormal data may come from equipment failure, network anomalies [7], or malicious attacks, which may have a serious impact on the stability and security of the system [8,9]. Through timely and accurate anomaly detection technology, we can quickly detect and handle these abnormal events, prevent potential security risks and system crashes, and ensure normal operation [10]. One of the methods for anomaly detection in the IoT is to collect real-time data from various devices, process and analyze temporal data using deep learning algorithms, and identify abnormal events that do not match normal behavior patterns. Generally speaking, IoT anomaly detection is performed through data collected by IoT sensors. They are mainly stored in the form of time series data. Time series data refers to data arranged in chronological order, which can reflect the trends and interrelationships of various factors in IoT systems [11].

Traditional anomaly detection methods include statistical, rule-based, and machine-learning methods. Statistical methods often set thresholds based on data distribution and determine whether values are abnormal by comparing them to the thresholds. Rule-based methods establish rules based on domain knowledge and historical data and determine whether values are abnormal by rules [12]. Machine learning methods distinguish normal and abnormal data through training learning algorithms. However, for each application scenario, the most suitable anomaly detection method needs to be chosen based on the characteristics of the data and practical requirements. With the development of artificial intelligence and deep learning technologies, neural network-based anomaly detection methods have gradually become a research hotspot. These methods utilize the learning ability and representation ability of neural networks to better capture complex patterns and features in time series data. For example, adaptive graph neural networks can be used to learn dynamic structures and dependencies in time series data, enabling more effective detection of anomalous events. In addition, deep learning-based anomaly detection methods can also utilize models such as convolutional neural networks (CNN) [13] or recurrent neural networks (RNN) [14] to extract features and classify time series data, identifying anomalous events. Except methods mentioned above, there are also other methods that can be applied to time series data [15]. For example, support vector machines (SVM) [16] can be trained on historical data to predict and classify future data; wavelet transform can decompose time series data into different frequency components to identify anomalous events; chaotic theory-based methods can utilize concepts and methods from chaotic theory to extract features and classify time series data.

Although supervised learning methods with labels often perform well in practice, their limitations are increasing. As the dimension increases, the relationship between sensors becomes more complex. As shown in Fig. 1, some sensor data has obvious temporal patterns and fluctuations, but some sensor data has no patterns at all. Compared to general time series data, multivariate IoT time series data often exhibits volatility, which manifests as changes and uncertainties between data. Uncertainties can be caused by various factors, such as changes in the natural environment (e.g., temperature, humidity, light, etc.), device performance variations (e.g., battery power, hardware malfunction, etc.), network latency, or packet loss. This volatility makes IoT data highly noisy and unstable, which increases the

difficulty of IoT anomaly detection. As the amount of data continues to raise, the difficulty and cost of data labeling also increase, making unsupervised learning methods increasingly popular. However, although unsupervised learning methods can handle unlabeled data, their performance is limited due to the lack of precise information about target anomalies. For this study, the main motivation is to accurately and robustly detect IoT time series anomalies. Currently, our model only detects anomalies. Our main research questions are: 1) How to leverage ML and DL models to achieve accurate anomaly detection for IoT time series? 2) How to enhance the robustness of anomaly detection in fluctuating IoT time series with limited labeled data? Specifically, IoT time series are collected from IoT devices or sensors, generally stored as multivariate time series. For example, the Secure Water Treatment (SWaT) dataset we used is an IoT dataset collected from a water treatment test bed. There are 51 sensors in SWaT, which record different information, respectively, such as water flows and water pressures, etc., stored in the form of a 51-variate time series. The proposed model is to analyze these data to detect anomalies in such fluctuating IoT multivariate time series. To address this problem, we propose an innovative deep learning model called the Adaptive Structural GCN-based Framework using Mean-Teacher Mechanism (AS-GCN-MTM). How to make full use of unlabeled data for semi-supervised training, thereby improving the generalization ability and performance of the model. The contributions of this paper are summarized as follows:

- We designed an adaptive graph structure learning module, which can learn the graph structure of graph data through neural network (converting the IoT time series into graph data), so that Graph Convolutional Networks (GCN) can better capture the spatio-temporal dependencies of IoT time series to achieve better anomaly detection.
- In order to make the best use of the rare labeled data, we introduce the Mean Teachers model to improve the robustness of the GCN model.
- To improve the inference speed and accuracy of GCN, we propose a key data augmentation technique based on graph structure learning.
- Through empirical experiments, we compare the proposed model with other state-of-the-art ones and validate the effectiveness of the proposed model on three open datasets.

The rest of the paper is organized as follows: Section 2 reviews the existing research. In Section 3, we present the details of the AS-GCN-MTM framework. In Section 4, we conduct experiments and analyze the results. Finally, we conclude and elaborate on future work in Section 5.

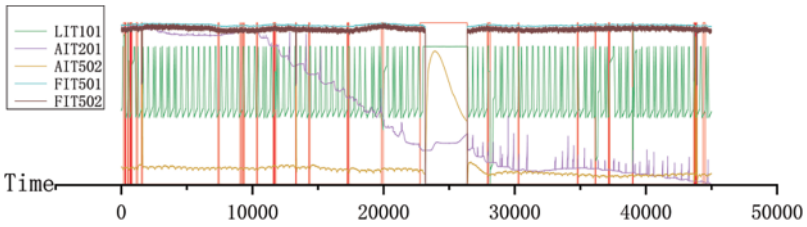

Figure 1: Display of the SWaT dataset, where the red parts are anomalies. LIT101, etc., denote sensors

2 Related Works

2.1 Methods Based on Machine Learning and Deep Learning

With the continuous development of technology and the increasing amount of data, modern time series analysis methods are focused on machine learning and deep learning technology for anomaly detection. Unlike traditional statistical methods, machine learning methods focus on

anomaly detection by learning patterns and features in the data rather than just inferring relationships between variables so that they can achieve higher accuracy and adaptability.

Machine learning methods are able to automatically extract useful features from large amounts of data and perform anomaly detection based on these features. This method can better deal with complex and changeable practical application scenarios and reduce the dependence on artificial feature selection. At the same time, deep learning technology further enhances anomaly detection performance and realizes deep learning by constructing neural network models. However, while machine learning and deep learning have achieved remarkable results in anomaly detection, each approach has its applicable scenarios and limitations. In practical applications, we need to consider the data characteristics, task requirements, and algorithm performance and choose the most suitable method to meet the specific anomaly detection requirements.

Common machine learning anomaly detection methods include K-Means [17] clustering, principal component analysis (PCA) [18], isolation forest (Isolation Forecast) [19], feature bagging [20], and so on. These methods each have their own principles and challenges, such as K-Means clustering requiring the selection of an appropriate number of clusters (K value) and isolation forest requiring the adjustment of window length parameters.

Common deep learning-based methods include variational autoencoder (VAE), generative adversarial network (GAN), adversarial training network AE (USAD) [21], long short-term memory network time series anomaly detection (LSTM-AD), stochastic recurrent neural network OmniAnomaly [22], etc. These methods can better capture complex patterns and features in time series data, but they also face different challenges, such as the blurry samples generated by VAE and the difficulties in training GAN [23,24]. However, each method has its own scope and limitations, and selecting the appropriate method requires a comprehensive consideration of data characteristics, task requirements, and algorithm characteristics.

2.2 Application of Graph Neural Network to Time Series

Graph neural networks (GNN) [25] play an increasingly important role in time series anomaly detection. They treat time series data as a graph structure, with each time point or data point as a node, and capture the relationships between nodes to mine patterns and anomalies in the data. Node relationships represent connections or interactions between data objects in a graph structure. This relationship can be directed or undirected and can have different weights. By analyzing these node relationships, we can gain insight into the pattern and structure of data objects [26]. In time series anomaly detection, the advantage of GNN is that they can capture the dynamic relationships and patterns. Time series data often exhibits complex dynamic patterns, such as trends, periodic changes, etc.; these patterns can be learned and represented by GNN. So that the aggregation operation of GNN can capture the relationships between nodes, it can better capture the complex patterns and anomalies in time series data [27].

GCN is able to capture both temporal and spatial relationships through the design of its graph structure, while time series data often contain temporal dependencies and spatial correlations. At present, many models have made active explorations and attempts to combine time series data with graph neural networks and have achieved certain results [28,29]. Graph Deviation Network (GDN) [30] is an anomaly detection method based on graph attention Network (GAT) [31]. It detects anomalies by calculating the difference between the feature representation of each node and the global average feature representation. GDN has achieved good results in anomaly detection of multivariate time series data. Multi-Task Graph Convolutional Network (MTGCN) [32] is a multi-task learning graph

convolutional network model. It improves the performance and generalization ability of the model by learning multiple tasks simultaneously, such as anomaly detection, time series prediction, etc. MTGCN also achieves good performance in anomaly detection of multivariate time series. Graph Convolutional Network for Temporal Data (ST-GCN) [33] is a graph convolutional network specifically designed for time series data. It captures dynamic changes in time series by introducing a temporal convolution layer and combining it with graph convolution operations. Temporal Anomaly Detection Graph Convolutional Network (TTADGCN) [34] is a graph convolutional network for anomaly detection in time series. It captures local patterns in time series by introducing one-dimensional convolution operations and uses graph convolution operations to capture relationships between nodes. This model can detect abnormal patterns in time series data and has good generalization ability. Temporal Stacked Generative Adversarial Network (TSGAN) [35] is a time series prediction model based on a generative adversarial network (GAN). It converts time series data into a graph structure and uses graph convolutional networks to capture the relationships between nodes.

3 Proposed Method

3.1 Main Framework

Our model is shown in Fig. 2. First, we input the data (A small amount of labeled and unlabeled IoT time series) into the graph structure learning module and obtain the relationship between the data (for example, for the IoT data is to obtain the relationship between sensors and sensors, the output result of the graph structure learning module is a weighted adjacency matrix, representing the mutual relationship between sensors and the sensitivity of sensors to anomalies). Then, we augment or weaken the original data according to the obtained adjacency matrix. For example, according to the adjacency matrix, there may be some sensors whose data and anomalies are very sensitive, so we will augment the data of these sensors. Our model AS-GCN-MTM consists of teacher models and student models with the same structure and different parameters. Cross entropy loss (Crit) is calculated by entering labeled data into the student model. Here we use binary cross entropy. The unlabeled data are entered into the student model and the teacher model, respectively, to obtain the results and calculate the mean square error loss MSE. The final Loss is calculated by weighting the previously calculated Crit and Mse. By minimizing the loss function, we can optimize the parameters of the model, which will gradually improve the prediction accuracy during the training process. Subsequently, the Student model uses the Loss to perform parameter updating, while the parameters of the Teacher model are smoothly moved and updated by the parameters of the Student.

3.2 Adaptive Graph Structure Learning

The graph structure of data is crucial for graph neural networks, as it provides the foundation for the training and inference of graph neural networks. Currently, the mainstream methods for obtaining the graph structure include node embedding, which maps each node in the graph to a low-dimensional vector representation, usually using linear transformation, nonlinear activation function, and normalization techniques. However, node embedding may not be able to fully capture the global information of the graph structure. Graph convolutional methods and graph embeddings can effectively capture the global information of the graph structure but require a large amount of computational resources and time. To address this problem, we propose an adaptive graph structure learning method based on neural networks. This method learns the graph structure through neural networks, which not only achieves good results but also consumes less time and resources. The purpose of adaptive graph structure learning is to use neural network science to learn and generate graph structures from input data.

$$M_1 = \tanh\left(E_1\left(Node_n\theta_1\right) * \alpha\right) \tag{1}$$

$$M_2 = \tanh\left(E_2\left(Node_n\theta_2\right) * \alpha\right) \tag{2}$$

$$A' = ReLU\left(a * \left(M_1 M_2^T - M_2 M_1^T\right)\right) \tag{3}$$

$$A = argtopk\left(A'\right) \tag{4}$$

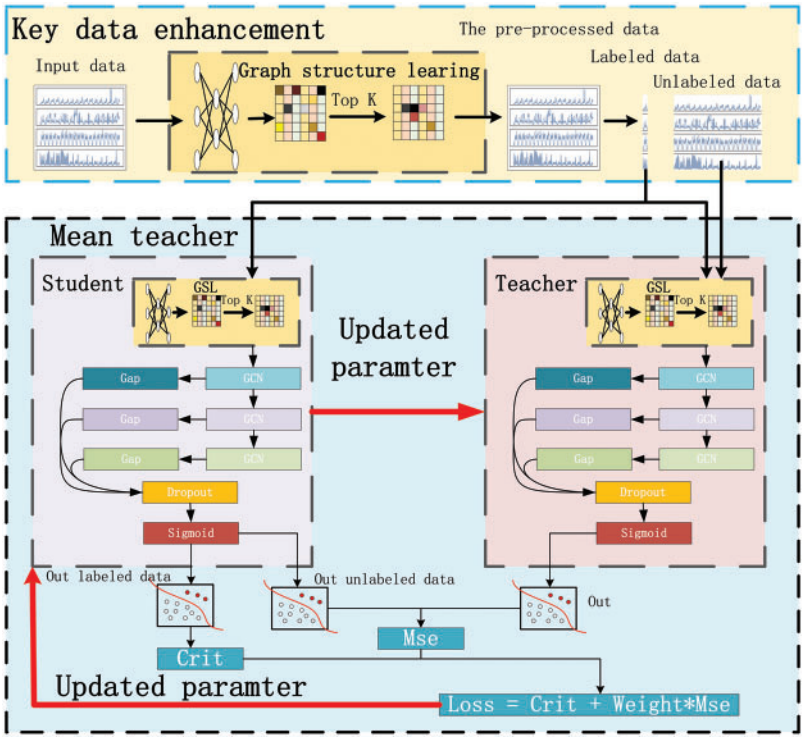

Figure 2: Main framework of the proposed method

As shown in Fig. 3, the graph structure learning layer first randomly initializes two matrices M_1 and M_2 [36] by vector embedding based on the number of nodes in the data. Through asymmetric transformation, it obtains the adjacency matrix A [37,38]. The adjacency matrix generated by graph structure learning represents the relationship between nodes, but not all relationships are important. We select the Top K relations with the highest weights. Where θ_1 and θ_2 represent the parameters of vector embedding, respectively, α and a represent the hyperparameters of network saturation. We generate matrices by vector embedding because we hope to achieve a self-optimizing and adaptive approach to obtaining graph structures through training the parameters of the embedding vectors. We perform asymmetric transformation because we believe that the relationships between sensors in reality are not necessarily symmetric. For example, in knowledge graphs, the relationships between entities are often asymmetric. Asymmetric graph structures can improve the efficiency and performance of the network. For some asymmetric relationships, if we use a symmetric network structure to handle them, it may waste computational resources and time because the relationship between two nodes only needs to pass information through one edge. We believe that not all node-to-node relationships are important. By including only important relationships, we can reduce the computational and storage requirements and improve efficiency and performance. We reduce computational and memory costs by sampling. As

in formula (3), we only compute asymmetric edges. The number of parameters utilized in the process of generating the graph structure in this manner is minimal. So, the principle of the graph structure learning layer is not complicated, and the computational cost is relatively low.

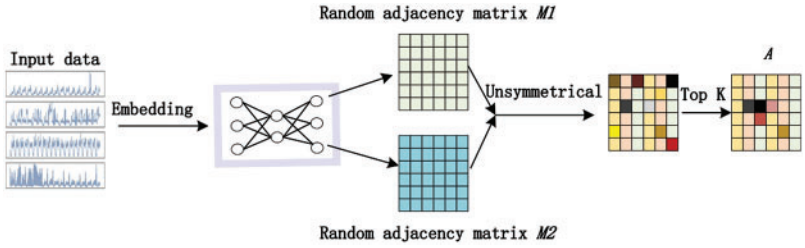

Figure 3: Graph structure learning

3.3 Key Data Augmentation Based on Graph Structure Learning

The multivariate time series data generated by the Internet of Things has high complexity, not only for its high data dimension but also for the complex interaction between different sensors. It makes it difficult to process and analyze these data. To solve this problem, we visualized the data and found that some sensor data are very sensitive to abnormal situations while others are not. Therefore, we augment the data of these sensitive sensors to improve our model's ability to identify and capture anomalies. First, we input the data into the graph structure learning module, by neural network to learn a weighted adjacency matrix that represents the relationship between the data. The weighted adjacency matrix in Fig. 4 represents the relationship between each sensor in datasets and the sensitivity of each sensor to anomalies. Based on the matrix, we can observe that some sensors are very sensitive to anomalies and some sensors have a strong connection between them. Next, we calculate the weights to select the top K sensors that are most sensitive to anomalies. Finally, we reprocess the selected top-K sensor data to improve our model and enhance its anomaly detection capabilities. In formula (5), X' indicates the routine normalization of the data, normalize the data. In formula (6), select K sensors that are most sensitive to anomalies. and formula (7) indicates the augmentation of the selected data.

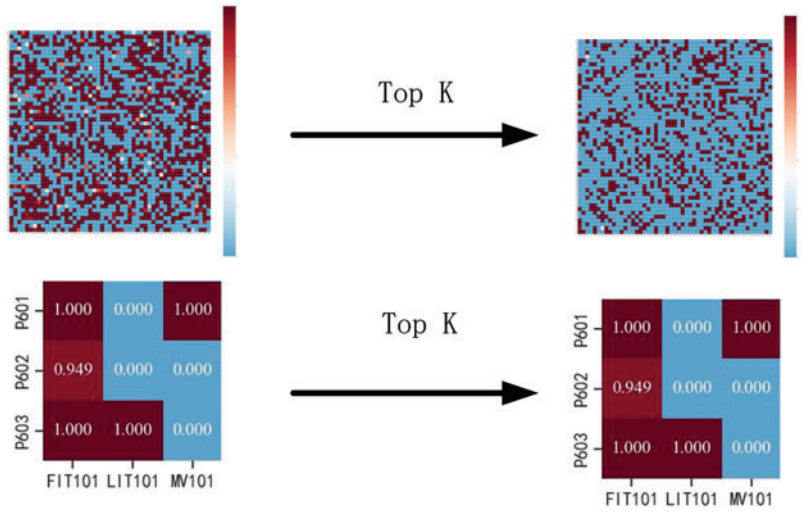

Figure 4: Relationship between sensors

$$X' = (X - X_min) / (X_max - X_min) \tag{5}$$

$$X' = argtopk\,(A[i, :]) \tag{6}$$

$$X' = \varepsilon X' \tag{7}$$

3.4 Mean Teacher Semi-Supervised Learning

Mean Teachers is an effective semi-supervised learning algorithm mainly used in natural language processing and computer vision. The core idea of the algorithm is to divide the model into teachers and students. Teachers are used to generate learning objectives for students, and students use the objectives provided by teachers to learn. It can improve the performance and generalization ability of the model using unlabeled data and achieve good results in some scenarios. However, it also has some shortcomings and limitations. It requires simultaneous processing of labeled and unlabeled data, which increases the computational and storage space requirements. Means Teachers can better adapt to this data sparsity, so that it can better analyze time series data.

$$L_{crit} = Crit\left(S_{X_L}, Label\right) \tag{8}$$

$$L_{mes} = Mes\,(S_X, T_X) \tag{9}$$

$$Loss = L_{crit} + \gamma L_{mes} \tag{10}$$

$$\theta'_t = \alpha\theta'_{t-1} + (1 - \alpha)\,\theta_t \tag{11}$$

S represents the student model, T represents the teacher model, X_L represents the labeled data, X represents the unlabeled data, S_{X_L} represents the student model input labeled data. The output result S_{X_L} is obtained through the student model and labeled data, and then the cross entropy loss Crit is calculated between S_{X_L} and the label L. The unlabeled data is then input into the student and teacher models to obtain S_X and T_X. Using S_X and T_X, we obtain the mean squared error loss MSE between them. The final loss is weighted by Crit and MSE [39], γ represents the weighted hyperparameter. The student model uses the final loss to optimize the parameters, while the teacher model's parameters θ'_t are updated by weighting the student model's parameters θ_t and the teacher model's temporal memory θ'_{t-1}.

4 Experiments

4.1 Datasets

As shown in Table 1, we used a total of three data sets to evaluate our model AS-GCN-MTM. These were two water datasets SWaT [40] and WADI [41], as well as a data set PSM collected from multiple application server nodes within eBay. (1) SWat is an international time series data set specifically for hydrological and environmental research. The data set contains many variables related to hydrology, such as rainfall, water level, flow, etc., as well as variables related to the environment, such as temperature, humidity, etc. (2) The WADI data set is characterized by high data quality and contains a variety of variables related to water resources. This makes the WADI data set a good data source for studying time series prediction and related issues. On the WADI data set, researchers can explore the effectiveness of various water management strategies and methods and apply them to practical water management problems. (3) The PSM (Pool Server Metrics) data set is a data set collected from multiple application server nodes within eBay. The source of the PSM data set is monitoring data collected from various nodes of the application server. These data are used to measure server performance and application health, including server resource usage, application response time, request throughput,

and more. For the performance evaluation of the model, we used Precision, Recall, and F1 scores as evaluation metrics for the model.

Table 1: Datasets and metrics

Datasets	Features	Train	Test	Animmalies
SWaT	51	36000	8992	12.2%
WADI	127	13824	3456	5.76%
PSM	25	24000	6000	32.5%

4.2 Experimental Settings and Baseline Model

We implemented our method and baseline model on Python 3.9, Pytorch version 1.13, and CUDA 11.6 on AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz and NVIDIA RTX 3070 graphics card. We compare the performance of our proposed method with eight popular anomaly detection methods. The purpose of this study is to compare various machine learning and deep learning methods to evaluate the performance of multiple methods in a more comprehensive way, provide references for subsequent research, and provide direction and ideas improving.

4.3 Evaluation Metrics

We adopt widely-used precision (Prec), recall (Rec), and F1 score (F1) as the evaluation metrics for our experiments. $Rec = TP/(TP + TN)$ represents the probability that the classification is correct in the total sample, and $Prec = TP/(TP + FP)$ represents the probability that the classification is actually correct in the sample. $F1 = (2 * Prec * Rec)/(Prec + Rec)$ as an evaluation index that can reflect both accuracy and recall rate. TP represents the number of positive cases that are correctly judged as positive cases, FN represents the number of positive cases that are incorrectly classified as negative cases, TN represents the number of negative cases that are correctly classified as negative cases, and FP represents the number of negative cases that are incorrectly classified as positive cases.

4.4 Experimental Result

The F1 score is an important indicator in evaluating anomaly detection models, which combines the accuracy and recall rate of the model and can reflect the overall performance of the model. In this table, we can observe that the performance of the AS-GCN-MTM model on the F1 score is generally better. On the three data sets, the F1 score of the AS-GCN-MTM model is higher than that of other models, with a data annotation rate of 0.1. The use case we envision is that using a small amount of labeling on the data will improve the performance of the model compared to unsupervised, and much less (<10%) labeled data is required, compared to supervised models. As demonstrated from the experimental result of all three datasets, the detection performance of our model increases with the rate of data labeling, as expected in typical semi-supervised models. Thus increased accuracy is even more evident in the WADI dataset compared to SWaT and PSM. The reason is that the WADI dataset has a dimensionality of 127, which is significantly higher than the other two datasets. Consequently, extracting features from the WADI dataset using unsupervised methods becomes challenging. As a result, data sets with high dimensionality and complex data like WADI typically exhibit low initial performance. However, as the rate of data annotation improves, there is a significant enhancement in model performance. On the other hand, both SWaT and PSM models show relatively modest

improvements due to their simpler nature and lower dimensionalities compared to WADI. Specifically, the SWaT dataset has 51 dimensions while the PSM dataset has 25 dimensions.

4.5 Result Analysis

Table 2 shows the performance of AS-GCN-MTM on three datasets with different data labeling rates. We can clearly see that the performance of AS-GCN-MTM improves with the increase in data labeling rates. In addition, on the SWaT dataset, the F1 score increased by 22% between AS-GCN-MTM-0.01 and AS-GCN-MTM-0.1, on the WADI dataset, the F1 score increased by 161% between AS-GCN-MTM-0.01 and AS-GCN-MTM-0.1, and on the PSM dataset, the F1 score increased by 42% between AS-GCN-MTM-0.01 and AS-GCN-MTM-0.1. For the SWaT and PSM datasets, although the F1 score of the model increases with the increase of the data labeling rates, the increase is relatively small. This may indicate that on these datasets, the increase of the data labeling rates has limited improvement on the model performance. On the WADI dataset, the increase of the F1 score exceeded 100%.

Table 2: The experimental results of AS-GCN-MTM and the baseline model on three datasets. AS-GCNMTM-0.01 and AS-GCN-MTM-0.02 indicate that the proportion of labeled data in the training dataset used by ASGCN-MTM is 1% and 2%, respectively

Datasets metrics	SWaT			WADI			PSM		
	R	F1	P	R	F1	P	R	F1	P
K-Means	0.171	0.184	0.199	0.495	0.373	0.3	0.141	0.216	0.463
PCA	0.686	0.763	0.86	0.445	0.318	0.247	0.126	0.193	0.418
FeB	0.154	0.165	0.178	0.19	0.153	0.128	0.9	0.1	0.053
VAE	0.706	0.715	0.726	0.475	0.335	0.259	0.143	0.218	0.454
USAD	0.915	0.812	0.73	0.81	0.634	0.519	0.06	0.114	0.993
MAD_GAN	0.764	0.674	0.602	0.584	0.549	0.519	0.103	0.187	**0.999**
OmniAnomaly	**0.999**	0.806	0.675	0.615	0.565	0.522	0.36	0.53	**0.999**
LSTM_AD	0.764	0.676	0.606	**0.81**	0.525	0.388	0.893	0.909	0.925
AS-GCN-MTM-0.01	0.603	0.729	**0.922**	0.29	0.33	0.389	0.638	0.676	0.718
AS-GCN-MTM-0.02	0.702	0.755	0.817	0.3	0.442	0.845	0.778	0.778	0.778
AS-GCN-MTM-0.04	0.736	0.799	0.874	0.375	0.519	0.842	0.846	0.869	0.892
AS-GCN-MTM-0.05	0.795	0.832	0.872	0.55	0.672	0.866	0.867	0.877	0.886
AS-GCN-MTM-0.06	0.797	0.835	0.876	0.695	0.716	0.739	0.891	0.91	0.93
AS-GCN-MTM-0.08	0.798	0.836	0.878	0.765	0.809	0.869	0.897	0.921	0.947
AS-GCN-MTM-0.1	0.846	**0.89**	0.938	0.84	**0.863**	**0.888**	**0.948**	**0.963**	0.978

Fig. 5 shows the performance comparison between AS-GCN-MTM and baseline models under different data label rates. Fig. 5 shows that the performance of AS-GCN-MTM-0.05 to AS-GCN-MTM-0.1 on the SWaT dataset is superior to all baseline models. On the WADI dataset, the performance of AS-GCN-MTM-0.05 to AS-GCN-MTM-0.1 is superior to all baseline models. On the PSM dataset, the performance of AS-GCN-MTM-0.06 to AS-GCN-MTM-0.1 is also superior to all baseline models, and the performance of AS-GCN-MTM-0.01 to AS-GCN-MTM-0.05 on the PSM dataset is only second to LSTM_AD. This indicates that the performance of the AS-GCN-MTM

model is generally better than the baseline model on the three datasets of SWaT, WADI, and PSM. The AS-GCN-MTM model has strong generalization ability and robustness, and it is able to effectively handle the task of anomaly detection on different datasets.

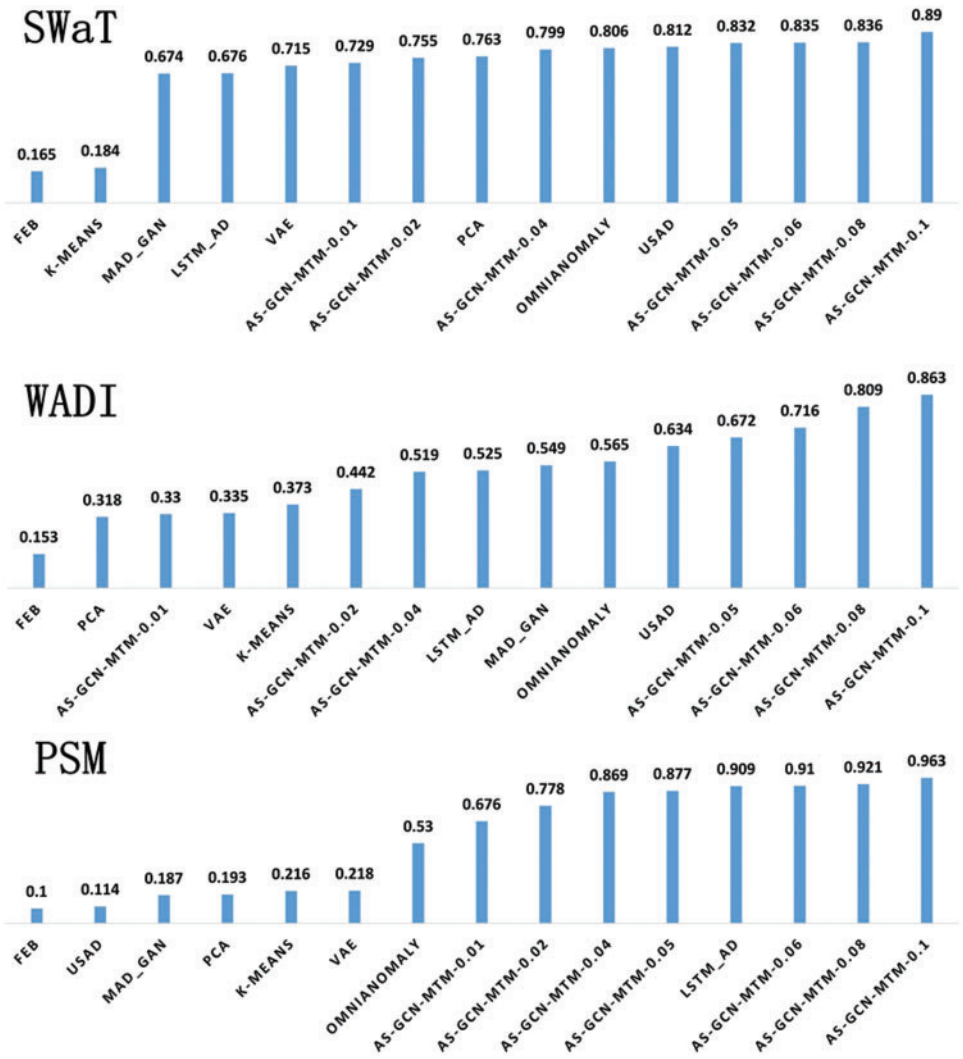

Figure 5: Comparison of baseline model performance with AS-GCN-MTM on three datasets, where the X-axis is the baseline method and AS-GCN-MTM with different data annotation rates, and they are sorted by F1 value

4.6 Ablation Experiments

In this section, we evaluate the effectiveness of graph structure learning and compare and analyze the impact of graph structure learning on the performance of AS-GCN-MTM on SWaT and WADI data sets, respectively. In the ablation experiment, we test the performance of model by moving the key data enhancement (kde) module and the graph structure learning (gls) module, respectively.

From the data in Tables 3 and 4, it is clear that graph structure learning has significantly improved the performance of AS-GCN-MTM on both data sets. Since key data augmentation is learned based

on graph structure, we did not conduct ablation experiments of key data augmentation alone. Although the average performance of critical data augmentation on SWaT datasets decreased slightly, we observed that adding critical data augmentation improved model performance at high data labeling rates. Both datasets show that key data augmentation can reduce the performance of the model when the tagging rate is low but can improve the performance when the tagging rate is high. On the WADI dataset, key data augmentation greatly improves the average performance of the model. After removing MeanTeacher, AS-GCN-MTM only utilizes 1%–10% of labeled data for supervised learning, and the findings indicate a significant decline in model performance across both datasets. Notably, the WADI dataset exhibits the most pronounced degradation in performance, with a decrease of up to 17.8%.

Table 3: Ablation experiments on SWaT datasets

Label	0.01	0.02	0.04	0.05	0.06	0.08	0.1	Average
AS-GCN-MTM	0.729	0.755	0.799	**0.832**	0.835	**0.836**	**0.89**	0.81
AS-GCN-MTM-kde	**0.733**	**0.767**	**0.83**	0.828	**0.839**	0.832	0.878	**0.815**
AS-GCN-MTM-gsl-kde	0.732	0.763	0.82	0.806	0.821	0.836	0.869	0.806
AS-GCN-MTM-mtm	0.736	0.753	0.815	0.802	0.818	0.822	0.869	0.802

Table 4: Ablation experiments on WADI datasets

Label	0.01	0.02	0.04	0.05	0.06	0.08	0.1	Average
AS-GCN-MTM	0.33	0.44	0.519	0.672	0.716	0.809	0.863	0.621
AS-GCN-MTM-kde	0.367	0.412	0.545	0.661	0.693	0.72	0.779	0.596
AS-GCN-MTM-gsl-kde	0.35	0.347	0.511	0.608	0.669	0.718	0.77	0.567
AS-GCN-MTM-mtm	0.297	0.385	0.463	0.581	0.509	0.632	0.717	0.51

4.7 Parametric Sensitivity Analysis

In this section, we evaluate the effect of parameters in the Mean Teacher model on model performance. We will then perform a sensitivity analysis on the learning rate parameter α of the student model without augmentation with key data. After applying augmentation with key data, we will conduct a sensitivity analysis on the number of GCN layers in the model and the value of γ in formula (10). The hyperparameters utilized in our model are as follows: α: 0.95, Number of GCN layers: 3, and γ: 0.2.

Through the analysis of parameter α in Table 5, we can see that the higher the parameter α is, the better the performance of AS-GCN-MTM is under the condition of high data annotation rate. Conversely, the lower the parameter α is, the better the performance of AS-GCN-MTM is under the condition of low data annotation rate. This indicates that in the case of high data annotation rate, the student model can learn more features of labeled data to achieve better performance, while in the case of low data annotation rate, the student model should reduce the learning rate of the teacher model. The optimal number of GCN layers and loss ratio hyperparameters are selected in our study.

Table 5: Hyperparameter sensitivity analysis

Label	α					layer			γ			
	0.99	0.95	0.8	0.65	0.5	2	3	4	0.2	0.4	0.6	0.8
0.1	0.871	0.878	0.848	0.864	0.854	0.88	0.89	0.879	0.89	0.878	0.873	0.859
0.08	0.839	0.832	0.815	0.825	0.827	0.819	0.836	0.808	0.836	0.802	0.806	0.816
0.06	0.843	0.839	0.829	0.839	0.833	0.815	0.835	0.811	0.835	0.821	0.805	0.824
0.05	0.822	0.828	0.83	0.832	0.835	0.803	0.832	0.784	0.832	0.812	0.804	0.782
0.04	0.822	0.83	0.829	0.819	0.83	0.814	0.799	0.822	0.799	0.814	0.823	0.836
0.02	0.756	0.767	0.761	0.771	0.78	0.737	0.755	0.748	0.755	0.767	0.734	0.743
0.01	0.732	0.733	0.732	0.729	0.735	0.729	0.729	0.729	0.729	0.728	0.726	0.729
Average	0.812	0.815	0.806	0.811	0.813	0.799	0.81	0.797	0.81	0.803	0.795	0.798

4.8 Real-Time Performance Analysis

In the context of the Internet of Things (IoT) environment, time series anomaly detection plays a critical role in ensuring system stability and prompt responsiveness. The data generated by IoT devices is typically characterized by continuity, high-speed transmission, and necessitates real-time processing to facilitate decision-making and response mechanisms. Consequently, it is imperative for the model to swiftly process data and deliver accurate results pertaining to anomaly detection within a limited timeframe. We measure the inference speed of the model under different configurations. For a 2 layer GCN, the average inference time per sample is approximately 0.25 ± 0.03 s; for a 3 layer GCN, it is around 0.33 ± 0.03 s; and for a 4 layer GCN, it amounts to about 0.71 ± 0.025 s.

5 Conclusion & Future Work

This study proposes a novel method for the Adaptive Structural GCN-based Framework using the Mean-Teacher Mechanism (AS-GCN-MTM) for Anomaly Identification in IoT system. This method aims to perform better than unsupervised methods using only a small amount of labeled data. Mean Teachers is an effective semi-supervised learning method that utilizes unlabeled data for training to improve the model's generalization ability. However, in time series data, the dependencies between data are often unknown, so traditional semi-supervised learning methods may not achieve optimal results. To address this issue, we design a neural network-based graph structure adaptive learning layer that can automatically learn the graph structure from time series data to better capture the relationships between nodes. Our model can more effectively detect anomalies and demonstrate superior experiment performance through this method. In future work, we plan to conduct more experiments and parameter sensitivity analysis to further verify the effectiveness of graph structure learning and mean teachers in our model. We will also try more model structures to integrate Mean Teachers better and improve model performance. Moreover, we will verify whether the proposed anomaly recognition method can effectively detect malicious interference, such as adversarial attacks. We will analyze the complexity and adaptability of the model in various IoT scenarios. Considering the diversity and dynamics of the IoT environment, we will gather additional datasets from different application scenarios, including smart homes, industrial monitoring, intelligent transportation, etc., to evaluate the model's performance across diverse scenarios. Simultaneously, we will explore methods to optimize the model structure to align with data characteristics and requirements specific to each scenario. This

includes enhancing graph construction techniques, improving time series data processing capabilities within the model, and designing more efficient algorithms to reduce computational complexity. Additionally, our focus will be on fortifying the model against adversarial attacks and malicious activities. Given the increasing prominence of IoT security issues, we aim to enhance its robustness against potential attacks such as data tampering and model spoofing by analyzing common attack vectors and devising corresponding defense strategies.

Acknowledgement: None.

Funding Statement: This research is partially supported by the National Natural Science Foundation of China under Grant No. 62376043 and Science and Technology Program of Sichuan Province under Grant Nos. 2020JDRC0067, 2023JDRC0087, and 24NSFTD0025.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Weijian Song; Xi Li; Jianhua Ren; Juan Chen; Peng Chen; Yunni Xia. Data collection: Xi Li; Jianhua Ren; Peng Chen. Analysis and interpretation of results: Weijian Song; Xi Li; Peng Chen. Draft manuscript preparation: Weijian Song; Xi Li; Juan Chen. Manuscript revision and supervision: Peng Chen, Yunni Xia. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: This paper mainly uses three datasets, SWaT, WADI, and PSM. Among them, SWaT and WADI are open datasets and available on request via https://itrust.sutd.edu.sg/ while PSM is available at https://github.com/eBay/RANSynCoders.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Sharma V, Sood D. Adoption of internet of things and services in the Indian insurance industry. In: Big data: A game changer for insurance industry. Leeds, UK: Emerald Publishing Limited; 2022. p. 35–42. doi:10.1108/978-1-80262-605-620221003.
2. Gao H, Wang X, Wei W, Al-Dulaimi A, Xu Y. Com-DDPG: task offloading based on multiagent reinforcement learning for information-communication-enhanced mobile edge computing in the internet of vehicles. IEEE Trans Vehicular Technol. 2024;73(1):348–61.
3. Pang G, Shen C, Cao L, Hengel AVD. Deep learning for anomaly detection: a review. ACM Comput Surveys. 2021;54(2):1–38. doi:10.1145/3439950.
4. Xin R, Chen P, Zhao Z. Causalrca: causal inference based precise fine-grained root cause localization for microservice applications. J Syst Softw. 2023;203:111724. doi:10.1016/j.jss.2023.111724.
5. Okey OD, Maidin SS, Adasme P, Lopes Rosa R, Saadi M, Carrillo Melgarejo D, et al. BoostedEnML: efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning. Sens. 2022;22(19):7409. doi:10.3390/s22197409.
6. Chen P, Xia Y, Pang S, Li J. A probabilistic model for performance analysis of cloud infrastructures. Concurrency Comput Pract Experience. 2015;27(17):4784–96. doi:10.1002/cpe.3462.
7. Song Y, Xin R, Chen P, Zhang R, Chen J, Zhao Z. Autonomous selection of the fault classification models for diagnosing microservice applications. Future Gener Comput Syst. 2024;153:326–39. doi:10.1016/j.future.2023.12.005.

8. Pan Y, Wang S, Wu L, Xia Y, Zheng W, Pang S, et al. A novel approach to scheduling workflows upon cloud resources with fluctuating performance. Mob Netw Appl. 2020;25:690–700. doi:10.1007/s11036-019-01450-0.
9. Long T, Chen P, Xia Y, Ma Y, Sun X, Zhao J, et al. A deep deterministic policy gradient-based method for enforcing service fault-tolerance in MEC. Chin J Electron. 2024;34:1–11. doi:10.23919/cje.2023.00.105.
10. Zhang R, Chen J, Song Y, Shan W, Chen P, Xia Y. An effective transformation-encoding-attention framework for multivariate time series anomaly detection in IoT environment. Mob Netw Appl. 2023:1–13. doi:10.1007/s11036-023-02204-9.
11. Xin R, Liu H, Chen P, Zhao Z. Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework. J Cloud Comput Adv Syst Appl. 2023;12(1):1–16. doi:10.1186/s13677-022-00383-6.
12. Gao H, Fang D, Xiao J, Hussain W, Kim JY. CAMRL: a joint method of channel attention and multidimensional regression loss for 3D object detection in automated vehicles. IEEE Trans Intell Transp Syst. 2023;24(8):8831–45. doi:10.1109/TITS.2022.3219474.
13. Kim P. Deep learning. Berkeley, CA: Apress; 2017. p. 103–20. doi:10.1007/978-1-4842-2845-6_5.
14. Jain LC, Medsker LR. Recurrent neural networks: design and applications. 1st ed. NW Boca Raton, USA: CRC Press; 1999.
15. Gao H, Wu Y, Xu Y, Li R, Jiang Z. Neural collaborative learning for user preference discovery from biased behavior sequences. IEEE Trans Comput Soc Syst. 2023:1–11. doi:10.1109/TCSS.2023.3268682.
16. Chu F, Jin G, Wang L. Cancer diagnosis and protein secondary structure prediction using support vector machines. Berlin, Germany: Springer Berlin Heidelberg; 2005; p. 343–63. doi:10.1007/10984697_16.
17. Krishna K, Murty MN. Genetic K-means algorithm. IEEE Trans Syst Man Cybernet Part B (Cybernet). 1999;29(3):433–9. doi:10.1109/3477.764879.
18. Maćkiewicz A, Ratajczak W. Principal components analysis (PCA). Comput Geosci. 1993;19(3):303–42. doi:10.1016/0098-3004(93)90090-R.
19. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. Pisa, Italy, IEEE; 2008. p. 413–22. doi:10.1109/ICDM.2008.17.
20. Lazarevic A, Kumar V. Feature bagging for outlier detection. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, USA; 2005; p. 157–66.
21. Audibert J, Michiardi P, Guyard F, Marti S, Zuluaga MA. USAD: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, USA; 2020. p. 3395–404.
22. Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Alaska, USA; 2019. p. 2828–37.
23. Chen P, Liu H, Xin R, Carval T, Zhao J, Xia Y, et al. Effectively detecting operational anomalies in large-scale IoT data infrastructures by using a GAN-based predictive model. Comput J. 2022;65(11):2909–25. doi:10.1093/comjnl/bxac085.
24. Qi S, Chen J, Chen P, Wen P, Niu X, Xu L. An efficient GAN-based predictive framework for multivariate time series anomaly prediction in cloud data centers. J Supercomput. 2023;80:1268–93. doi:10.1007/s11227-023-05534-3.
25. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y, et al. Graph attention networks. Stat. 2017;1050(20). doi:10.48550/arXiv.1710.10903.
26. Chen Z, Ge Z. Knowledge automation through graph mining, convolution, and explanation framework: a soft sensor practice. IEEE Trans Ind Inf. 2021;18(9):6068–78. doi:10.1109/TII.2021.3127204.

27. Song Y, Xin R, Chen P, Zhang R, Chen J, Zhao Z. Identifying performance anomalies in fluctuating cloud environments: a robust correlative-GNN-based explainable approach. Future Gener Comput Syst. 2023;145:77–86. doi:10.1016/j.future.2023.03.020.
28. Xing M, Ding W, Zhang T, Li H. STCGCN: a spatio-temporal complete graph convolutional network for remaining useful life prediction of power transformer. Int J Web Inf Syst. 2023;19(2):102–17. doi:10.1108/IJWIS-02-2023-0023.
29. Hofer D, Jäger M, Mohamed AKYS, Küng J. A study on time models in graph databases for security log analysis. Int J Web Inf Syst. 2021;17(5):427–48. doi:10.1108/IJWIS-03-2021-0023.
30. Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2021. p. 4027–35. doi:10.1609/aaai.v35i5.16523.
31. Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? arXiv preprint arXiv: 181000826. 2018.
32. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C. Connecting the dots: multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020. p. 753–63.
33. Song C, Lin Y, Guo S, Wan H. Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2020. p. 914–21.
34. Deng L, Lian D, Huang Z, Chen E. Graph convolutional adversarial networks for spatiotemporal anomaly detection. IEEE Trans Neural Netw Learn Syst. 2022;33(6):2416–28. doi:10.1109/TNNLS.2021.3136171.
35. Gao N, Xue H, Shao W, Zhao S, Qin KK, Prabowo A, et al. Generative adversarial networks for spatio-temporal data: a survey. ACM Trans Intell Syst Technol. 2022;13(2):1–25. doi:10.1145/3474838.
36. Hamilton WL, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Neural information processing systems. California, USA: Curran Associates Inc.; 2017.
37. Liu Z, Sun M, Zhou T, Huang G, Darrell T. Rethinking the value of network pruning. arXiv preprint arXiv: 181005270. 2018.
38. Vu QH, Ooi BC, Papadias D, Tung AK. A graph method for keyword-based selection of the top-k databases. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, CA; 2008. p. 915–26.
39. Klinker F. Exponential moving average versus moving exponential average. Math Semesterberichte. 2011;58:97–107. doi:10.1007/s00591-010-0080-8.
40. Mathur AP, Tippenhauer NO. SWaT: a water treatment testbed for research and training on ICS security. In: 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), 2016; Vienna, Austria. p. 31–6. doi:10.1109/CySWater.2016.7469060.
41. Ahmed CM, Palleti VR, Mathur AP. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In: Proceedings of the 3rd International Workshop on Cyber-physical Systems for Smart Water Networks, Pennsylvania, USA; 2017; p. 25–8.