A Novel Semi-supervised IoT Time Series Anomaly Detection Model Using Graph Structure Learning

Weijian Song[1], Peng Chen[1](✉), Juan Chen[1](✉), Yunni Xia[2], Xi Li[1],
Qinghui Xi[1], and Hongxia He[1]

[1] School of Computer and Software Engineering, XiHua University, Chengdu, China
{chenpeng,chenjuan}@mail.xhu.edu.cn
[2] School of Computer Science, Chongqing University, Chongqing, China

Abstract. Internet of Things (IoT) is an evolving paradigm for building smart cross-industry. The data gathered from IoT devices may have anomalies or other errors for various reasons, such as malicious activities or sensor failures. Anomaly detection is thus in high need for guaranteeing trustworthy execution of IoT applications. Existing IoT anomaly detection methods are usually built upon unsupervised methods and thus can be inadequate when facing complex IoT data regularity. In this article, we propose a semi-supervised approach for detecting IoT time series anomalies based on Graph Structure Learning (GSL) using multi-layer perceptron Graph Convolutional Networks (GCN) and the Mean Teachers (MT) mechanism. The proposed model is capable of leveraging a small amount of labeled data (1% to 10%) to achieve high detection accuracy. We adopt Mean Teachers to utilize unlabeled data for enhancing the model's detection performance. Moreover, we design a novel graph structure learning layer to adaptively capture the IoT data features among different nodes. Experimental results clearly suggest that the proposed model outperforms its competitors on two public IoT datasets, achieving 82.85% in terms of F1 score and 22.8% increase.

Keywords: IoT Time Series · Anomaly Detection · Graph Structure Learning · Graph Convolutional Networks · Semi-supervised · Mean Teachers

1 Introduction

The Internet of Things (IoT) is an emerging means that consists of collaborative terminals and sensors connected through the Internet. The IoT can be applied in different application domains, such as smart homes, wearable devices, smart

This research is supported by the National Natural Science Foundation under Grant No. 62376043 and Science and Technology Program of Sichuan Province under Grant No. 2020JDRC0067, No. 2023JDRC0087, and No. 2020YFG032662376043, and Chunhui Project of Ministry of Education of China under Grant No. Z2011085.

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 562, pp. 375–391, 2024.
https://doi.org/10.1007/978-3-031-54528-3_21

cities, healthcare, agriculture, transportation, and industry. The major strength of the IoT is that it helps to make appropriate decisions based on the data collected by sensors, and tracks devices in a smart way.

A typical IoT environment involves a large number of interconnected sensors, such as those found in water treatment plants, power plants, or transportation systems. Generally, real-time data collected from these sensors are processed and stored as IoT multivariate time series. The variables in these time series are often interrelated; for example, in a water treatment plant, if sensor data monitoring water flow rate exhibits anomalies, the sensor data monitoring water pressure is also likely to show abnormalities. Due to the a mass of sensors and the complexity of their relationships, performing anomaly detection in more intricate and large-scale IoT systems is generally more difficult.

IoT anomaly detection [1] holds significant importance and value in modern society. IoT devices are typically distributed across various geographical locations, monitoring a large amount of device status and operational data aids in predicting potential failures or damages By continuously monitoring anomalies in real-time, it becomes possible to forecast potential device failures in advance and carry out timely maintenance, thereby reducing maintenance costs and downtime, while enhancing equipment reliability and availability [2]. IoT systems involve extensive data transmission and processing, often operating in resource constrained environments (such as sensor nodes, embedded devices, etc.) [3,4]. Anomaly detection can help identify abnormal data flows, energy consumption, and more, optimizing resource allocation and improving system efficiency and performance [5]. IoT anomaly detection not only ensures the stability, security, and efficiency of IoT systems but also enhances user experience, providing robust support for data analysis and intelligent decision-making. As such, it holds crucial significance and value in IoT applications. Figure 1 depicts an example of IoT time series anomaly detection, where the highlighted red portion indicates the detected anomaly. Notably, methods for anomaly detection in IoT data have undergone extensive research and development, resulting in widespread exploration.

The methods for anomaly detection have evolved from classical approaches initially to machine learning and deep learning methods in recent years, achieving significant advancements. For instance, as early as 1979, Tukey [5] introduced a statistical method for detecting anomalies in time series. In the past decade, machine learning and deep learning methods have achieved tremendous success in computer vision tasks, leading researchers to apply these approaches to anomaly detection. For example, Autoregressive model (VAR) [6], Long Short-Term Memory (LSTM) [7], Variational Autoencoder (VAE) [8], Generative Adversarial Networks (GAN) [9]. Graph Neural Networks (GNNs) have attracted considerable attention in the realm of anomaly detection for time-series data [10]. Their ability to capture relationships and dynamic changes within time-series data has led to superior performance. They are capable of capturing relationships and dynamic changes in time-series data. GNNs can be mainly divided into Convolutional Graph Neural Networks (GCN), Graph Attention

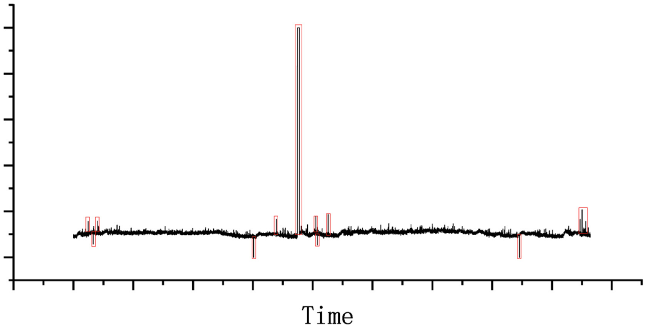

Fig. 1. A typical anomaly detection scenario of IoT time series.

Networks (GAT) [11], and Graph Neural Networks with gated updates, among others.

Despite the achievements mentioned above, when it comes to multi-dimensional time series anomaly detection in the context of IoT, there are still the following difficulties and challenges: (1) The detection accuracy of unsupervised methods remains insufficient; (2) Extracting and representing precise spatiotemporal data features from complex multi-dimensional IoT time series data is still a challenge.

To address such problems, we propose the semi-supervised Mean Teachers [12] based Graph Convolutional Network Model for IoT time series anomaly detection (MTGCN). The main contributions are summarized as follows:

- To address the challenge of data labeling difficulty, we introduce the Mean Teachers model, which enables leveraging unlabeled data for semi-supervised training, thereby enhancing the model's generalization ability and performance.
- To improve the detection accuracy of the model, we employ a multi-layer perceptron graph convolutional network (GCN) based on adaptive graph structure learning as the foundational framework. Compared to traditional distance-defined graph structures, this adaptive graph structure learning method enables us to acquire superior graph relationships, thereby boosting the performance of our model.
- The experimental results demonstrate that this approach still outperforms the majority of unsupervised methods in anomaly detection on two publicly collected real-world datasets, even when using a very small amount (1%–10%) of labeled data.

The rest of the paper is organized as follows. Section 2 reviews the existing research on anomaly detection based. In Sect. 3, we propose the semi-supervised model based on GSL and give a detailed description of each module. In Sect. 4, we conduct experiments and provide experimental results and analysis. Finally, we conclude and elaborate on future work in Sect. 5.

2 Related Work

We briefly review anomaly detection methods for time series, including both classical approaches and those based on machine learning and deep learning.

2.1 Classic Methods

Anomaly detection [13] is the task of finding abnormal data in the data. The detection of anomalies in time-series data has always held a crucial position in the field of anomaly detection. Many classical methods for anomaly detection are based on statistical techniques, while in recent years, numerous scholars have developed machine learning-based approaches for anomaly detection. Subsequently, methods based on deep neural networks have also become increasingly popular.

There are many traditional methods for time series anomaly detection, such as AR (AutoRegressive), MA (Moving Average), and ARMA (AutoRegressive Moving Average). Autoregressive model (AR) [14] is one of the fundamental models for univariate time series and is a linear model. AR model predicts a variable's future values by regressing on its own past values, assuming that the relationship between past and future values is consistent over time. The anomaly score is determined by the difference between the predicted value and the observed value [15]. The AR model is a classic statistical method used in time series anomaly detection. However, the AR model assumes that the data is stationary, so using AR for time series anomaly detection requires certain data requirements or necessitates necessary data preprocessing. The Moving Average model (MA) and AutoRegressive model (AR) are both linear models, but they differ in that AR uses past observed values as differences, while MA uses past residual errors as differences. The AutoRegressive Moving Average model (ARMA) is a combination of AR and MA and is commonly used for univariate time series.

2.2 Methods Based on Machine Learning and Deep Learning

ML-Based Methods. Different from statistical methods, the purpose of using machine learning methods for anomaly detection is to make the most accurate predictions or detections by inferring relationships between variables. Currently, there are many popular machine learning anomaly detection methods such as K-Means [16] clustering, Principal Component Analysis (PCA) [17], Isolation Forest, Feature Bagging, and more. K-Means is one of the classical clustering methods used for anomaly detection in machine learning. It calculates the distance between targets based on Euclidean distance [18]. The principle is to divide the sample set into K clusters based on the distances between samples. The goal is to have points within the same cluster as close together as possible and points from different clusters as far apart as possible. Principal Component Analysis (PCA) is a common data analysis technique often used for dimensionality reduction in high-dimensional data. It is also employed for anomaly detection by extracting the main feature components of the data. The primary steps of PCA for anomaly detection involve reducing the dimensions and then calculating the differences between the vectors obtained after dimensionality reduction and the original vectors. Isolation Forest [19] is an anomaly detection algorithm that isolates anomalies by constructing binary trees and measuring the number of steps required to isolate data points from the majority of the dataset.

DL-Based Methods. Based on deep learning, various methods have gained significant popularity recently, such as Variational Autoencoders (VAE) [20], Generative Adversarial Networks (GAN) [21–23], Unsupervised Adversarial Training of Autoencoders (USAD) [24], LSTM-based Time Series Anomaly Detection (LSTM-AD), and OmniAnomaly using Random Recursive Neural Networks [25]. The Variational Autoencoder (VAE) compresses input data into a code through an encoder and then decodes the code back into the input through a decoder. Through continuous learning, the output becomes increasingly similar to the input. VAE can learn the latent variables in the data, allowing it to generate entirely new samples rather than simply replicating the input data. The anomaly score of VAE is determined by the difference between the input (original data) and the output (reconstructed data). The primary challenge of VAE is that the generated samples can often be blurry or less precise. This is because the generation process is random, and VAE cannot guarantee that every generated sample is of high quality.

Generative Adversarial Networks (GANs) train the generator and the discriminator in an adversarial manner, ultimately making it difficult for the discriminator to distinguish the data generated by the generator. The challenge of GANs lies in achieving Nash equilibrium during training, where sometimes it can be accomplished using gradient descent, while in other cases, it may be difficult to achieve. The Unsupervised Adversarial Training of Autoencoders (USAD) combines the advantages of both autoencoders (AE) and Generative Adversarial Networks (GAN). It achieves this by continuously training two AE networks in an adversarial manner. Long Short-Term Memory Network for Time Series Anomaly Detection (LSTM-AD) utilizes LSTM, which is a type of Recurrent Neural Network (RNN) architecture. Compared to Convolutional Neural Networks (CNN), data in LSTM flows only forward, making it a type of feedforward neural network. A major issue with RNNs is the problem of vanishing gradients, which LSTM addresses by using gated units. LSTM-AD typically involves making predictions using Long Short-Term Memory (LSTM) networks and then computing the prediction errors to detect anomalies. LSTM can learn time dependencies, but learning long-term dependencies in lengthy time series can be quite challenging. OmniAnomaly employs Random Recursive Networks and flat normalization to generate reconstruction probabilities. This method outperforms many deep learning approaches, but its training time is relatively large.

The application of GNN in time series anomaly detection has been gradually attracting attention as they can capture relationships and dynamic changes between sequences. GNNs can be used to construct time-dependent graph structures and detect anomalous behavior by learning representations of nodes and edges in time series.

Graph Neural Networks (GNN) use network embedding to represent network nodes as low-dimensional vectors while preserving the network topology and node information. They then perform subsequent tasks such as classification, clustering, etc. For instance, Graph Convolutional Networks (GCN) aggregate neighboring nodes' features to represent the node's characteristics. Other classic graph neural networks include Graph Attention Networks (GAT), Spatio-Temporal Graph Convolutional Networks (ST-GCN) [26], and more. GNNs have found significant applications in time series anomaly detection. For instance, the Graph Deviation Network (GDN) [27], based on GAT, has achieved excellent results in anomaly detection for IoT multivariate time series. MTGNN [28], which uses GCN, also exhibits remarkable performance in anomaly detection for multivariate time series.

3 Proposed Framework

3.1 Architecture

The structure of MTGCN is illustrated in Fig. 2, where both the Student and Teacher models share identical architectures. Initially, input data is divided based on their labeling into labeled and unlabeled data. The training data for the Student model consists of both labeled and unlabeled data, while the Teacher model only receives unlabeled data. The graph structure learning module is trained and updated in conjunction with the training of the Student model, continuously updating the neural network's parameters. For the Student model, the labeled data is processed through multiple layers of Graph Convolutional Networks (GCNs) to obtain output results, which are then compared with the data labels to compute the cross-entropy loss (Cross-Entropy Loss (Crit)). Additionally, the unlabeled data processed by the Student model's GCNs is compared with the unlabeled data processed by the Teacher model's GCNs to compute the mean squared error (Mean Squared Error Loss (MSE)). Finally, the Student model's parameters are optimized based on the combined loss (weighted sum of Crit and MSE), while the Teacher model updates its parameters using the parameters of the Student model.

3.2 Graph Structure Learning

Our framework for anomaly detection follows a process where, firstly, the Internet of Things (IoT) dataset is transformed into graph-structured data. Next, we use Graph Convolutional Networks (GCN) to learn the relationships between sensors in this data. Finally, we employ the learned GCN for anomaly detection (Fig. 3).

In GCN, the spatial dependency between nodes is represented by the adjacency matrix A. However, existing methods for constructing an adjacency matrix

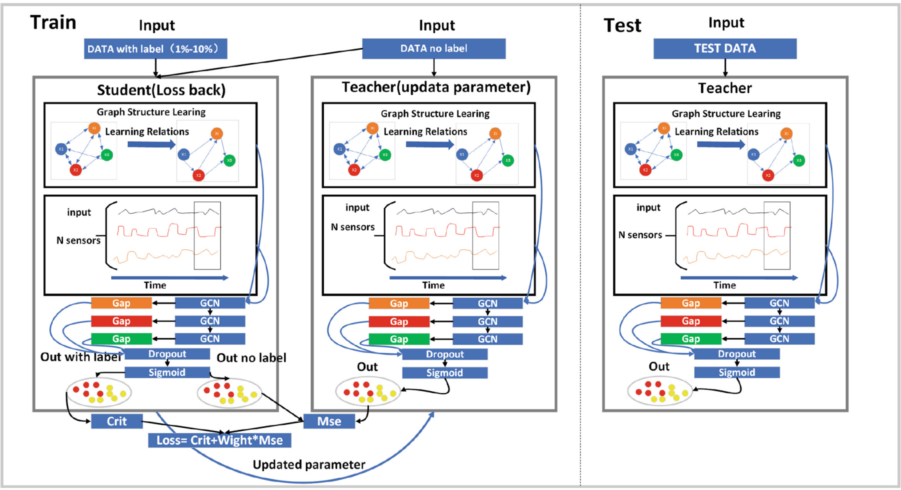

Fig. 2. Overview

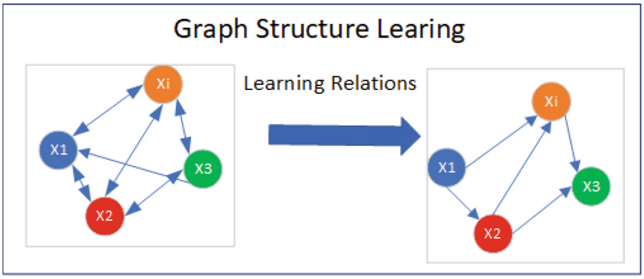

Fig. 3. Graph Structure Learning

for a graph typically involve computing the similarity between nodes using distance metrics like Euclidean distance. This approach can be computationally and spatially expensive, especially for large graphs, and the effectiveness of the adjacency matrix constructed through distance computation may not always be optimal. To address these limitations, we propose a neural network-based approach to learn the adjacency matrix [29].

$$M_1 = \tanh\left(Embedding_1\left(Node_n\theta_1\right) * \alpha\right) \tag{1}$$

$$M_2 = \tanh\left(Embedding_2\left(Node_n\theta_2\right) * \alpha\right) \tag{2}$$

$$A = Relu\left(\alpha * \left(M_1 M_2^T - M_2 M_1^T\right)\right) \tag{3}$$

$$For\ i = 1, 2, 3 \cdots n \tag{4}$$

$$I = argtopk\left(A\ \left[i, :\ \right]\right) \tag{5}$$

$$A\left[i, -I\ \right] = 0 \tag{6}$$

M_1 and M_2 are matrices with randomly initialized parameters. Where $Embedding_1$ and $Embedding_2$ represent randomly initialized node embeddings, and $Node_n$ represents the number of nodes in the graph. θ_1 and θ_2 are model parameters, and α is a hyperparameter representing network saturation. As the relationships between sensors may not be symmetric, our adjacency matrix is transformed non-symmetrically using Formula 3. i represents the top i edges with the highest selected weights. Finally, we utilize the *argtopk* operation to sparsify the adjacency matrix, selecting the top K edges with the highest correlations to obtain the final adjacency matrix A [30,31].

3.3 Mean Teacher Semi-supervised Learning

Mean Teacher is an effective semi-supervised learning method that fully utilizes unlabeled data to improve model performance. It performs exceptionally well in scenarios with limited labeled data. In our model, we leverage the Mean Teacher semi-supervised approach for training and detection, where a small amount (1%–10%) of training data is labeled. The idea behind Mean Teacher is that the model serves as both a teacher and a student. The teacher model is a replica of the student network with the same architecture, and its parameters are exponentially averaged from the student network. The student model learns using the targets generated by the teacher model, and its parameters are continually updated during training to adapt to the data.

$$Loss_{crit} \ = Crit\left(S_{X_L}\ ,\ L\right) \tag{7}$$

$$Loss_{mes} = Mes\left(S_X, T_X\right) \tag{8}$$

$$Loss = Loss_{crit} + \beta Loss_{mes} \tag{9}$$

$$\theta_t^{'} = \alpha\theta_{t-1}^{'} + (1-\alpha)\,\theta_t \tag{10}$$

S_{X_L} represents the output of the student model for labeled data, L represents the labels of the labeled data, S_X and T_X represent the outputs of the student and teacher models for unlabeled data, respectively. The specific steps are as follows: input the data, both labeled and unlabeled, into the student model to obtain S_{X_L}, S_X, and T_X outputs. Calculate the loss terms: $Loss_{crit}$ and $Loss_{mes}$ [32]. Finally, update the parameters of the student model based on the combined loss using Formula 9. Then, use formula 10 to update the parameters of the teacher model, leveraging the parameters of the updated student model.

4 Experiments

4.1 Datasets

In this paper, we utilized two sensor datasets based on a water treatment physical testbed system: SWat (Secure Water Treatment) and WADI (Water Distribution). In both datasets, operators simulated scenarios where real-world water treatment plants were subjected to attacks, and the recorded anomalies represent genuine occurrences. The SWat dataset originates from a water treatment testbed coordinated by the Public Utilities Board of Singapore (Mathur and Tippenhauer [33]). It consists of six interlinked processes, forming a representation of a small-scale IoT system mirroring real-world scenarios. In this study, the SWat dataset comprises data from 51 sensors, with anomalies accounting for 12.2% of the data. On the other hand, the WADI dataset is an extension of SWat, consisting of data from 127 sensors in total.

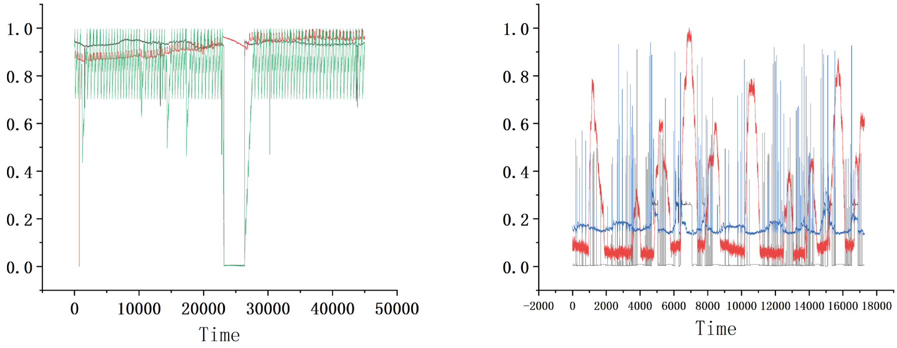

Fig. 4. The left (a) shows the feature representation of the WADI dataset, and the right (b) shows the feature representation of the SWaT dataset.

Figure 4 depict the feature representations of the two datasets, revealing significant differences in the feature distributions between them (Table 1).

Table 1. Datasets

Datasets	Features	Train	Test	Anomaly Rate
SWaT	51	36000	8992	12.2%
WADI	127	13824	3456	5.76%

The types of anomalies in time series data mainly include point anomalies, contextual anomalies, long-term trend anomalies, seasonal anomalies, cyclic anomalies, and so on. This paper focuses on the detection of point anomalies. Point anomalies are one of the most common anomaly types in time series data and are typically caused by sudden, unusual events, or errors. Point anomalies can lead to one or more data points deviating significantly from the normal data pattern. These anomalies can have a significant impact on businesses or systems, so timely detection and handling are essential.

4.2 Evaluation Metrics

We adopt widely-used precision (Prec), recall (Rec), and F1-Score (F1) as the evaluation metrics for our experiments.

4.3 Experimental Setup

We implement our method in PyTorch version 1.13 with CUDA 11.6 and PyTorch Geometric Library version 2.2.0, and train them on a server with AMD Ryzen 7 5800H with Radeon Graphics @ 3.20 GHz and NVIDIA RTX 3070 graphics cards.

We compared machine learning and deep learning methods in our study. For machine learning, the methods included K-Means and PCA. As for deep learning, the compared methods were VAE, USAD, LSTM-AD, MAD_GAN, and OmniAnomaly based on Random Recursive Neural Network.

4.4 Experimental Studies

"MTGCN-0.1", "MTGCN-0.08"... in Tables 2 and 3 respectively represent the MTGCN model at different data annotation rates, with bold numbers indicating the maximum value in that column. In the SWaT dataset, the highest F1 score is achieved by MTGCN with a 10% data annotation rate, reaching 87.8%. The highest Recall (Rec) score is attained by OmniAnomaly, reaching 99.9%, while the highest Precision (Prec) score is achieved by MTGCN with a 1% data annotation rate, reaching 99.8%. In Table 3, with a 10% data annotation rate, MTGCN achieves a 77.9% F1 score. The highest Recall scores are obtained by LSTM_AD and USAD, reaching 8%. The highest Precision score is achieved by MTGCN with a 10% data annotation rate, reaching 82%.

4.5 Result Analysis

Tables 2 and 3 provide the precision (Prec), recall (Rec), and F1-Score (F1) of the MTGCN and baseline models on the SWaT dataset and WADI dataset. All the baseline models and MTGCN perform worse on the WADI dataset compared to the SWaT dataset. We compared classical machine learning methods and deep learning methods, where the machine learning methods included K-Means, PCA, and FeB, while the deep learning methods consisted of VAE, USAD, MAD GAN, OmniAnomaly, and LSTM AD.

MTGCN achieves significantly higher F1-Scores (F1) at 10% data labeling rate on both datasets compared to the state-of-the-art baseline models. Among the baseline models, USAD performs the best on both datasets, achieving an F1 of 0.812 on the SWaT dataset and an F1 of 0.634 on the WADI dataset. MTGCN outperforms the state-of-the-art baseline models by 8.1% on the SWaT dataset and 22.8% on the WADI dataset at 10% data labeling rate. Furthermore, MTGCN's performance on the SWaT dataset is 2.2% higher than the state-of-the-art baseline models at a 4% data labeling rate, and on the WADI dataset, it is 4.2% higher at a 5% data labeling rate.

Table 2. Experimental Results on the SWaT Dataset.

Data	Method	Rec	F1	Prec
SWat	K-Means	0.495	0.373	0.3
	PCA	0.445	0.318	0.247
	FeB	0.19	0.153	0.128
	VAE	0.475	0.335	0.259
	USAD	0.915	0.812	0.73
	MAD_GAN	0.764	0.674	0.602
	OmniAnomaly	**0.999**	0.806	0.675
	LSTM_AD	0.764	0.676	0.606
	MTGCN-0.1	0.837	**0.878**	0.923
	MTGCN-0.08	0.781	0.832	0.891
	MTGCN-0.06	0.773	0.839	0.918
	MTGCN-0.05	0.764	0.828	0.904
	MTGCN-0.04	0.775	0.83	0.893
	MTGCN-0.02	0.684	0.767	0.876
	MTGCN-0.01	0.583	0.733	**0.988**

Table 3. Experimental Results on the WADI Dataset.

Data	Method	Rec	F1	Prec
WADI	K-Means	0.495	0.373	0.3
	PCA	0.445	0.318	0.247
	FB	0.19	0.153	0.128
	VAE	0.475	0.335	0.259
	USAD	**0.81**	0.634	0.519
	MAD_GAN	0.584	0.549	0.519
	OmniAnomaly	0.615	0.565	0.522
	LSTM_AD	**0.81**	0.525	0.388
	MTGCN-0.1	0.744	**0.779**	**0.82**
	MTGCN-0.08	0.68	0.72	0.767
	MTGCN-0.06	0.648	0.693	0.748
	MTGCN-0.05	0.625	0.661	0.705
	MTGCN-0.04	0.433	0.545	0.739
	MTGCN-0.02	0.295	0.412	0.7
	MTGCN-0.01	0.288	0.367	0.518

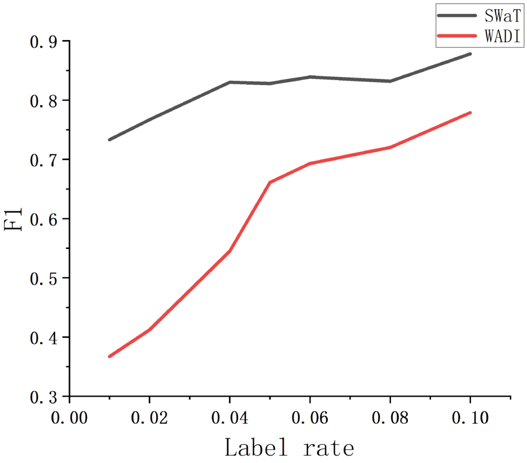

Fig. 5. MTGCN performance on two datasets.

Figure 5 displays the experimental results of MTGCN on the SWaT dataset and WADI dataset at different data labeling rates. Overall, the F1 scores of MTGCN on both datasets decrease as the data labeling rate decreases. Notably, MTGCN performs better on the SWaT dataset compared to the WADI dataset. This suggests that MTGCN is more effective in utilizing labeled data on the SWaT dataset, resulting in higher F1 scores, even as the amount of labeled data decreases. However, on the WADI dataset, MTGCN's performance suffers more when the data labeling rate is reduced. This difference in performance between the two datasets indicates that the characteristics and challenges of the datasets may play a role in influencing MTGCN's effectiveness under limited labeled data conditions.

From Fig. 6, Among the various models evaluated on the SWat dataset, we observe that MTGCN consistently demonstrates strong performance across different levels of data annotation, with F1 scores ranging from 0.878 at a 10% annotation rate to 0.733 at a 1% annotation rate. This consistent high performance suggests that MTGCN is particularly robust and effective in anomaly detection tasks with varying levels of labeled data. The decreasing trend in F1 score as the annotation rate decreases is less pronounced for MTGCN compared to other models in the list. This indicates that MTGCN has a notable advantage in handling scenarios with limited labeled data, making it a promising choice for anomaly detection tasks in real-world situations where labeled data may be scarce or expensive to obtain.

Figure 7 illustrates the comparison between MTGCN and the baseline models on the WADI dataset. From Fig. 7, it can be observed that at a data labeling rate of 5%, MTGCN outperforms all baseline models. Additionally, at a data labeling rate of 4%, MTGCN's performance is only surpassed by the USAD, MAD_GAN, and OmniAnomaly baseline models.

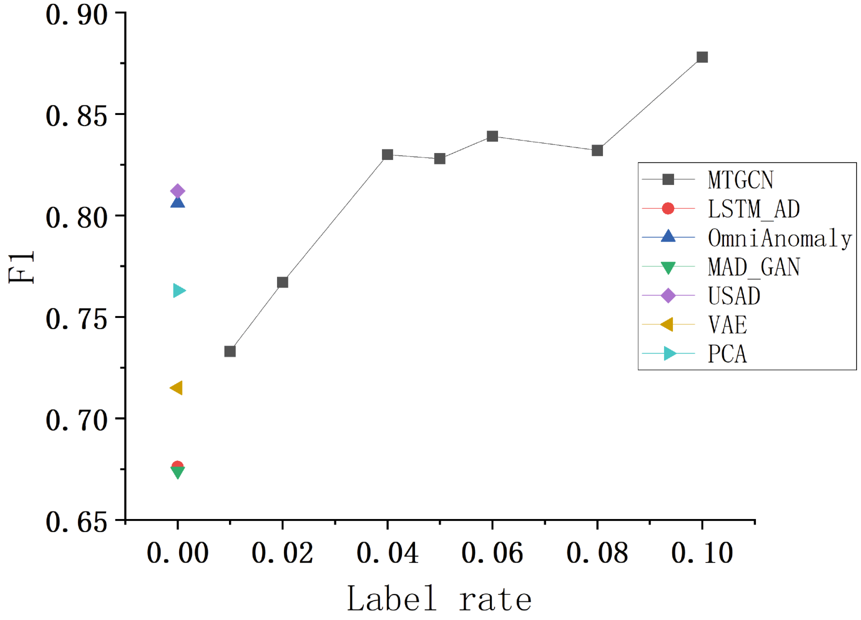

Fig. 6. Comparison of MTGCN on SWaT data set and baseline model.

4.6 Ablation Study

To validate the effectiveness of Graph Structure Learning (GSL), we conducted ablation experiments and observed the changes in F1 scores on both datasets (Fig. 8).

Table 4. Results of GSL ablation experiments on SWaT datasets.

label	0.1	0.08	0.06	0.05	0.04	0.02	0.01
MTGCN + GSL	0.878	0.832	0.839	0.828	0.83	0.767	0.733
MTGCN	0.869	0.836	0.821	0.806	0.82	0.763	0.732

From the data analysis in Table 4, it can be concluded that MTGCN, when using Graph Structure Learning (GSL), shows an average performance improvement of 1% on the SWaT dataset across all data labeling rates. The highest performance improvement, reaching up to 2.7%, is observed when the data labeling rate is 5%. From the data analysis in Table 5, it can be concluded that MTGCN, when using Graph Structure Learning (GSL), exhibits an average performance improvement of 4.5% on the WADI dataset across all data labeling rates. The highest performance improvement, reaching up to 8.7%, is observed when the data labeling rate is 5%.

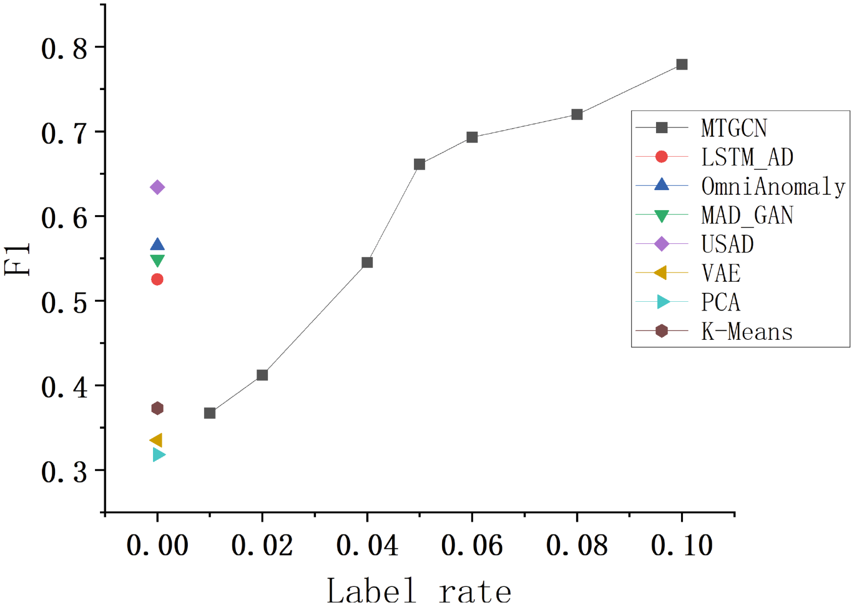

Fig. 7. Comparison of MTGCN on WADI data set and baseline model.

Table 5. Results of GSL ablation experiments on WADI datasets.

label	0.1	0.08	0.06	0.05	0.04	0.02	0.01
MTGCN + GSL	0.779	0.72	0.693	0.661	0.545	0.412	0.367
MTGCN	0.77	0.718	0.669	0.608	0.511	0.347	0.35

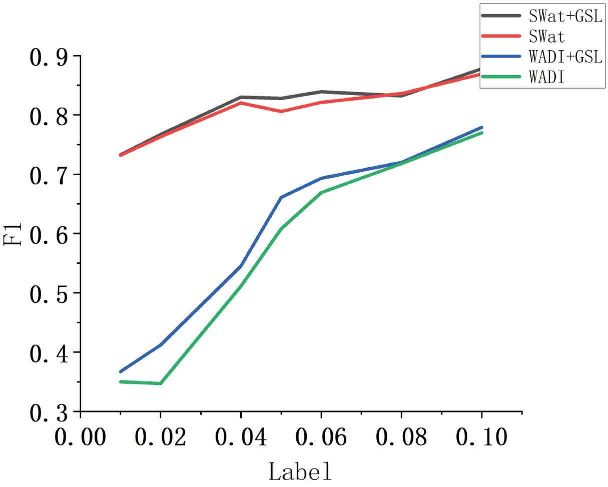

Fig. 8. Comparison of Ablation Experiments for GSL Learning on Two Datasets.

5 Conclusion and Future Work

MTGCN has demonstrated its effectiveness in anomaly detection on both the SWaT dataset and WADI dataset. Through a series of experiments and analyses, the introduction of Graph Structure Learning (GSL) in the MTGCN model has shown significant benefits. On the SWaT dataset, MTGCN outperforms all baseline models at a data labeling rate of 4%. Notably, at a lower data labeling rate of 2%, MTGCN's performance is only surpassed by the USAD and OmniAnomaly baseline models. The experimental results reveal that MTGCN effectively leverages the Graph Structure Learning (GSL) component, resulting in an average performance improvement of 1% on the SWaT dataset across all data labeling rates. The maximum performance gain of 2.7% is achieved at a data labeling rate of 5%. This indicates that the introduction of graph structure information significantly enhances MTGCN's anomaly detection capabilities on the SWaT dataset. Similarly, on the WADI dataset, MTGCN exhibits significant performance improvements with the inclusion of Graph Structure Learning (GSL). With an average performance gain of 4.5% across all data labeling rates, MTGCN consistently outperforms the baseline models. At a data labeling rate of 5%, MTGCN achieves its highest performance boost of 8.7%, demonstrating its superiority over the baseline methods. These findings emphasize the importance of Graph Structure Learning (GSL) in enhancing MTGCN's anomaly detection capabilities on both datasets. MTGCN effectively utilizes graph structure information, enabling it to adapt and excel even with limited labeled data, making it a competitive and effective approach in practical anomaly detection scenarios. The experimental results validate MTGCN's superiority over baseline models and underscore the significant role of Graph Structure Learning (GSL) in improving anomaly detection performance, establishing MTGCN as a competitive and effective method in anomaly detection research.

In future research, we intend to attempt more datasets and explore alternative methods to adapt to Mean Teachers

References

1. Pang, G., et al.: Deep learning for anomaly detection: a review. ACM Comput. Surv. (CSUR) **54**(2), 1–38 (2021)
2. Sharma, B., Sharma, L., Lal, C.: Anomaly detection techniques using deep learning in IoT: a survey. In: 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), pp. 146–149. IEEE (2019)
3. Chen, P., et al.: A probabilistic model for performance analysis of cloud infrastructures. Concurr. Comput. Pract. Exp. **27**(17), 4784–4796 (2015)
4. Pan, Y., et al.: A novel approach to scheduling workflows upon cloud resources with fluctuating performance. Mob. Netw. Appl. **25**, 690–700 (2020)
5. Tukey, J.W.: Exploratory Data Analysis, vol. 2 (1977)
6. van den Oord, A., et al.: WaveNet: a generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)

7. Filonov, P., Lavrentyev, A., Vorontsov, A.: Multivariate industrial time series with cyber-attack simulation: fault detection using an LSTM-based predictive data model. arXiv preprint arXiv:1612.06676 (2016)
8. Bodin, E., et al.: Nonparametric inference for auto-encoding variational Bayes. arXiv preprint arXiv:1712.06536 (2017)
9. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Veličković, P., et al.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
12. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
13. Braei, M., Wagner, S.: Anomaly detection in univariate time-series: a survey on the state-of-the-art. arXiv preprint arXiv:2004.00433 (2020)
14. Bali, T.G., Mo, H., Tang, Y.: The role of autoregressive conditional skewness and kurtosis in the estimation of conditional VaR. J. Bank. Financ. **32**(2), 269–282 (2008)
15. Chandola, V.: Anomaly detection for symbolic sequences and time series data. University of Minnesota (2009)
16. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS, vol. 2431, pp. 15–27. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45681-3_2
17. Shyu, M.-L., et al.: A novel anomaly detection scheme based on principal component classifier. In: Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, pp. 172–179. IEEE Press (2003)
18. Li, Y., et al.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
19. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. IEEE (2008)
20. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013)
21. Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, S.-K.: MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) ICANN 2019. LNCS, vol. 11730, pp. 703–716. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30490-4_56
22. Chen, P., et al.: Effectively detecting operational anomalies in large-scale IoT data infrastructures by using a GAN-based predictive model. Comput. J. **65**(11), 2909–2925 (2022)
23. Qi, S., et al.: An efficient GAN-based predictive framework for multivariate time series anomaly prediction in cloud data centers. J. Supercomput. **80**, 1–26 (2023)
24. Audibert, J., et al.: USAD: unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3395–3404 (2020)
25. Su, Y., et al.: 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, pp. 2828–2837 (2019)

26. Nicolicioiu, A., Duta, I., Leordeanu, M.: Recurrent space-time graph neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
27. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5 (2021)
28. Wu, Z., et al.: 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, pp. 753–763 (2020)
29. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural Information Processing Systems, vol. 30 (2017)
30. Liu, Z., et al.: Rethinking the value of network pruning. arXiv preprint arXiv:1810.05270 (2018)
31. Vu, Q.H., et al.: 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, pp. 915–926 (2008)
32. Klinker, F.: Exponential moving average versus moving exponential average. Math. Semesterber. **58**, 97–107 (2011)
33. Mathur, A.P., Tippenhauer, N.O.: 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), pp. 31–36. IEEE (2016)