Time Series Anomaly Detection Model Based on Memory-enhanced Transformer and Graph Network Joint Training

Qingqing Luo
School of Software
Xinjiang University
Urumqi, Xinjiang, China
estellalqq@gmail.com

Jiangang Dong*
School of Software
Xinjiang University
Urumqi, Xinjiang, China
137384753@qq.com

Abstract

To address the challenges of overgeneralization and inter-variable interactions inherent in multivariate time series data, this study proposes an unsupervised anomaly detection model, METG, which integrates a memory-enhanced Transformer with graph joint training. The memory enhancement module captures and records diverse patterns between normal data and memory items, significantly improving the model's representation capabilities for time series. Simultaneously, a graph convolutional network extracts spatial dependencies and feature interactions by leveraging graph structures. By combining the strengths of temporal and spatial modeling through a joint optimization objective function and anomaly detection criteria, METG achieves robust performance. Experimental evaluations on real-world datasets across four domains demonstrate the model's effectiveness, achieving an average F1 score of 94.5%.

CCS Concepts

• **Theory of computation** → Theory and algorithms for application domains; Machine learning theory.

Keywords

Unsupervised anomaly detection, Time series, Graph neural network, Memory network, Transformer

ACM Reference Format:
Qingqing Luo and Jiangang Dong. 2024. Time Series Anomaly Detection Model Based on Memory-enhanced Transformer and Graph Network Joint Training. In *2024 4th International Conference on Computational Modeling, Simulation and Data Analysis (CMSDA 2024), December 06–08, 2024, Hangzhou, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3727993.3728054

1 Introduction

Time series anomaly detection is an important research topic in data mining and is widely used in industrial equipment operation and maintenance. Efficient and accurate anomaly detection helps companies to continuously monitor key metrics and provide timely warnings of potential events [1]. Anomalous events are extremely rare and difficult to characterize compared to normal time-series data, which is usually easily accessible. Collecting sufficient labelled training data is not only time-consuming and laborious, but even if it is successfully obtained, the severe quantitative imbalance between normal and abnormal data still affects the model training effectiveness. Therefore, unsupervised detection techniques have become a practical approach to address this real-world problem.

Unsupervised time series anomaly detection has seen significant advances, aiming to learn general patterns from normal training data to effectively detect anomalies in unseen data. Recent research has proliferated in this area, with self-encoders (AEs) becoming a prominent unsupervised technique. AEs effectively minimize reconstruction error for normal data, using this error to identify anomalies. Building on the AE framework, methods such as LSTM-AE[4], Convolutional Auto-Encoder (CAE)[2,3], and ConvLSTM-AE[5] have been developed, proving effective in anomaly detection. Variational Auto-Encoder (VAE)[6] further improves on AE by learning the representation of the data in the latent space to enhance the robustness of the decoder to better deal with the noise in the input data. The InterFusion model[7], proposed by Li et al., learns the temporal and spatial dependent features of the data simultaneously by layering the VAE with two latent variables. Generative Adversarial Networks (GANs)[8], on the other hand, generate high-quality samples similar to real data through a game of generators and discriminators. Recently, Abdulaal et al.[9] in their study generated simultaneous representations by spectral analyses of the latent representations of the data for better detection of anomalies in multivariate data. The MST-GAT[10] model further explores temporal and spatial correlations by combining multimodal graph attention networks and temporal convolutional networks.

While significant progress has been made, unsupervised anomaly detection still faces two key challenges. First, the generalization ability of models is limited by the scarcity of normal training data, particularly with complex time series. Existing models often fail to capture the full diversity of normal patterns, impacting their generalization performance. Second, modeling the correlations between multivariate time series remains challenging. The complex dependencies among variables are difficult for current methods to capture, affecting detection accuracy.

To address the above challenges, this paper proposes a multivariate time series detection model (METG) based on the joint training of memory-enhanced Transformer and graph neural networks. The

*Corresponding author.

CMSDA 2024, Hangzhou, China
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1183-1/2024/12
https://doi.org/10.1145/3727993.3728054

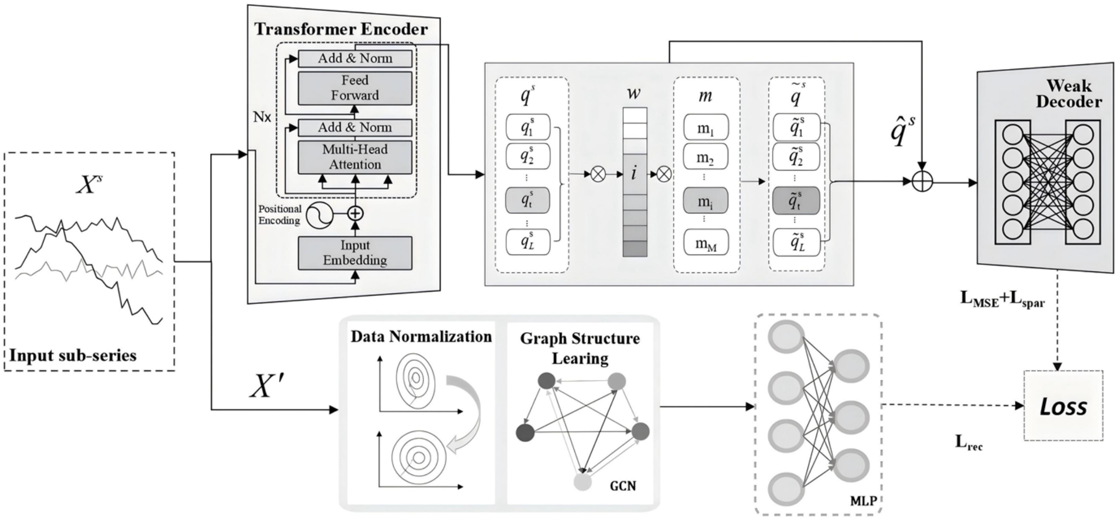

Figure 1: METG structure.

model strengthens the modelling capability of temporal and spatial features, and takes into account the correlation between different time series variables while improving the generalization capability. The main contributions of this paper are as follows:

- We propose the METG framework, which integrates memory-enhanced Transformers and graph neural networks, achieving excellent performance on four public datasets.
- Integration of memory networks into Transformer enhances the characterization capability by recording various patterns between normal data and memory items, which improves the time-series data processing capability and better distinguishes between normal and anomalous data.
- We introduce graph convolution to model spatial relationships, capturing dependencies between different locations in multivariate time series.
- By jointly optimizing the objective function, we propose an anomaly detection criterion that considers both anomaly scores and the importance of memory items, leading to improved overall performance.

2 Approach

2.1 Problem statement

A multivariate time series can be represented as $X = \{X^1, \cdots, X^N\}$, N represents the total number of subsequences. Each subsequence $X^s = [x_1^s, \cdots, x_L^s] \in \mathbb{R}^{L\times n}$consists of data points of length L and dimension n. The corresponding labels of the time series data are denoted as $Y = \{y_1, \cdots, y_N\}$, where $y_i \in \{1, \cdots, K\}$represents the category labels of the subsequence and K represents the number of categories of the data.

For long time series, a fixed length input sequence is generated through a sliding window of length n. The task of multivariate time series anomaly detection is to generate an output vector y, $y_i \in \{0, 1\}$which indicates whether the subsequence corresponding to the i-th timestamp is anomalous or not.

2.2 Basic Module

Figure 1 illustrates the overall architecture of METG, which mainly consists of an encoder-decoder architecture with a memory enhancement module and a graph convolutional network. The encoder-decoder part first inputs the input sub-sequenceX^sinto the Transformer encoder and the output features of the encoder are used as a query $q_t \in \mathbb{R}^C (t = 1, \cdots, L)$ and stored in the memory module. The input to the decoder is the updated query $\hat{q}_t \in \mathbb{R}^C (t = 1, \cdots, L)$,which consists of the combined features of the query $\tilde{q}_t$updated by the memory module and the original queryq_t.The decoder then maps the updated query back to the input space and outputs the reconstructed input subsequence $\hat{X}^S \in \mathbb{R}^{L\times \mathrm{n}}$. The GCN constructs a graph structure based on the input subsequence, where each univariate time series is treated as a feature. An adjacency matrix is created from the nearest neighbors of the feature vectors to capture both local similarity and global structure, providing meaningful inputs to the graph neural network. These two components are jointly optimized via a shared objective function to effectively detect anomalies in both temporal and spatial dimensions.

2.2.1 Encoders and decoders. Transformer[11] exhibits encoder-decoder excellence in sequence modelling as compared to VAE-based approaches. Based on this, various Transformer variants further enhance anomaly detection. For example, Anomaly Transformer[12] introduces an anomaly-awareness mechanism to better distinguish between normal and anomalous behaviors in both univariate and multivariate time series. In the METG model proposed in this paper, Transformer is combined with a memory network to map input subsequences into the latent space, capturing temporal dependencies. The Transformer's ability to capture temporal features is strengthened by learning and storing multiple patterns of normal data in the memory module. The decoder consists of two fully connected layers that weakly decode the input.

2.2.2 Memory networks. Recurrent Neural Networks (RNNs) and LSTM-based models capture long-term dependencies in sequences through local memory units, but these memory modules often become unstable over time. To address this, memory networks introduce a long-term memory module that can be stored and updated. MemAE[13] was the first to integrate a memory network into a self-encoder architecture, allowing the storage of different patterns of normal data in memory items. Specifically, the memory module is instantiated as a matrix $M \in \mathbb{R}^{F\times C}$that stores F dimensional C memory items. Each memory item$m_i (i = 1, \cdots, F)$ represents a different normal mode.

Given a query q_t, which represents a single query vector at moment *t*. The memory enhancement module updates the query by matching the probabilities between the weights w and the memory items $m_i$to output $\tilde{q}_t$. The relationship between the specific query $\tilde{q}_t$ and the memory matrix M is as follows:

$$\tilde{q}_t = wM = \sum_{i=1}^{C} w_i m_i \tag{1}$$

Where the *i*-th element w_i of the weight vector w is calculated based on the normalized similarity between the queryq_t and the memory itemm_i:

$$w_i = \frac{\exp(< m_i, q_t > /\tau)}{\sum_{j=1}^{M} \exp(< m_j, q_t > /\tau)} \tag{2}$$

Where t denotes the temperature hyperparameter. The query $q_t$is connected to $\tilde{q}_t$updated with memory terms along the feature dimensions to form the updated query $\hat{q}_t$.

$$\hat{q}^s = Concat\left(q^s, \tilde{q}^s\right) \tag{3}$$

The updated query is used as the new input to the decoder. During training, the memory matrix is updated to record normal features via the reconstruction loss function. Anomalous patterns tend to be ”cancelled out” by the normal patterns stored in the memory, resulting in reconstructed outputs of anomalous data that closely resemble those of normal samples. This makes anomaly reconstruction more challenging, improving the model's ability to distinguish between normal and anomalous data, while preventing overfitting.

2.2.3 Graph convolutional networks. Graph Convolutional Networks (GCN)[14] is a deep learning model designed for graph-structured data. GCN efficiently learns node feature representations by propagating information between nodes and their neighbors. In the METG model, GCN is applied to multivariate time series, treating each time series as a complete graph where nodes represent features, and edges capture the relationships between connected features.

A directed graph can be represented as$G = (V, E)$, where V denotes the nodes in the graph and E denotes the set of edges in the graph. The adjacency matrix $A \in \mathbb{R}^{N\times N}$of the graph represents the connectivity between the nodes in the graph, $A_{ij} = 1$ denotes the existence of an edge between node *i* and node *j* and vice versa$A_{ij} = 0$. The GCN consists of several graph convolutional layers, which are then updated by a linear transformation and a nonlinear activation function. The specific process is shown in Figure 2

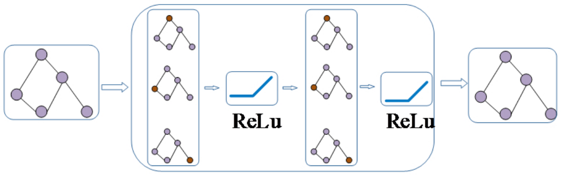

Figure 2: Graph Convolutional Network Architectural Framework.

The METG model incorporates graph convolution to enhance spatial modeling, feeding raw input sequences into the graph convolution module for joint characterization. The key steps are as follows:

a) Data normalization

To improve robustness, each time series is normalized based on the training data to eliminate magnitude differences across variables. Specifically, max-min normalization is applied as shown in Equation 4:

$$X^{'} = \frac{X^s - \min(D_{train})}{\max(D_{train}) - \min(D_{train})} \tag{4}$$

b) Graph Structure Learning

Relationships between node variables are learned by embedding, encoding variable relationships as edges of a directed graph G. For each node *i*, its first *k* neighbors are learned to obtain a sparse adjacency matrix A. The parameter *k* controls the sparsity of the generated neighborhood graph G.

$$e_{ij} = \alpha \cdot \frac{v_i - v_j}{||v_j|| \cdot ||v_i||} \tag{5}$$

$$A_{ij} = \ell\{j \in TopK(\{e_{ki} : k \in C_i\})\} \tag{6}$$

Where,$C_i \subseteq \{1, 2, \cdots, N\}$, e_{ij} is the cosine similarity between any two variable nodes *i* and *j*. TopK(·) returns the top k neighbors. C_i value is the value of the candidate relationship specified by the expert, and all nodes are assumed to be connected to each other when the prior relationship is unknown.

2.3 Joint Optimization Training

As previously stated, Transformer and GCN each have their own strengths that complement each other. METG combines the two to capture the data distribution of a time series through joint training. During training, both models' parameters are updated simultaneously. Algorithm 1 summarizes the main steps of METG model training.

The loss function of METG is defined as the sum of multiple optimization objective functions, and the overall METG training objective function is:

$$Loss = L_{MSE} + \lambda_1 L_{\text{rec}} + \lambda_2 L_{spar} \tag{7}$$

Where$L_{MSE}$and$L_{rec}$are the memory-enhanced Transformer model loss function and the reconstruction-based GCN model loss function, respectively:

$$L_{MSE} = \sum_{i=1}^{N} ||f_d\left(concat\left(q_i, \tilde{q}_i\right)\right) - X_i||_2^2 \tag{8}$$

Algorithm 1 Training algorithm for METG

```
Input: Input subsequence$X^s$, $f_e$(encoder), $f_d$(decoder),
hyperparameter$\lambda_1$,$\lambda_2$,$W \in \mathbb{R}^C$.
Output: The trained model.
$q^s = f_e(X^s)$.
Initialising the memory module: $c = K - means(q^{rand})$ // $q^{rand} \in \mathbb{R}^{N_{rand} \times L \times n}$ Randomly selected samples sequence.
for $i = 1,...,M$ do
    $\mathrm{m_i} = \mathrm{c_i}$.
return $m$.
$\hat{q}^s = Concat(q^s, w^s m)$.
$\hat{X}^s = f_d(\hat{q}^s)$.
$\widehat{q}^s = GCN(q^s)$.
$L = \sum_{i=1}^{N} ||f_d(concat(q_i, \tilde{q}_i)) - X_i||_2^2 + \lambda_1 \sum_{i=1}^{N} ||X_i - \tilde{X}_i||_2^2 + \lambda_2 \sum_{i=1}^{C} -w_i \cdot \log(w_i)$.
```

$$L_{rec} = \sum_{i=1}^{N} ||X_i - \tilde{X}_i||_2^2 \tag{9}$$

To avoid complex combinations of memory entries leading to excessive reconstruction anomalies, we use a sparse loss function L_{spar}that minimizes the entropy of the memory matrix weights w:

$$L_{spar} = \sum_{i=1}^{C} -w_i \cdot \log(w_i) \tag{10}$$

2.4 Anomaly Detection

In the METG model, the calculation of the anomaly score is closely related to the joint optimization objective function. The anomaly score incorporates the advantages of each model and aims to maximize the overall anomaly detection effect. Specifically, this paper adopts the following methods for the calculation of the anomaly score:

$$Score\left(X^s\right) = m\_score * (L_{MSE} + L_{rec}) \tag{11}$$

For each branch (Transformer-based and GCN-based), its corresponding anomaly scores L_{MSE} and $L_{rec}$are computed, respectively. at the same time, considering the correlation between the memory item and the overall data pattern, a memory item-related weight score m_{score} is introduced. this weight score reflects the importance of the memory item to the query result, and occupies a certain weight in the final anomaly judgement.

$$m_{score} = soft\max\left(||\tilde{q}_t - m_i||_2^2\right) \tag{12}$$

The final anomaly score is derived by multiplying the sum of the model scores by the weight scores. After training, the anomaly score is calculated for each timestamp, and a threshold σ is applied to determine if a data point is anomalous. The thresholds are dynamically selected using the Peak Over Threshold (POT-K)[15] method.

3 Experiments

3.1 Datasets and indicators

We evaluate the performance of METG on multivariate time series datasets in four different domains:

(a) Server Machine Dataset (SMD)[16]: This includes metrics such as CPU utilization, network utilization, and memory utilization of servers.

(b) Mars Science Laboratory rover (MSL)[17]: This data comes from sensors and actuators of NASA's Mars rover.

(c) Soil Moisture Active Passive satellite (SMAP)[17]: information on soil samples provided by NASA.

(d)Pooled Server Metrics (PSM)[9]: server performance data collected internally by multiple eBay application server nodes.

Table 1 shows the details of the dataset, including the number of samples in the Training, Validation and Test sets, P (%) represents the anomaly rate, and Dim represents the data dimension size of the dataset.

$P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$, $F1 = 2 \cdot \frac{P \cdot R}{P+R}$, where P, R denote precision and recall, respectively, TP is true positive, FP is false positive and FN is false negative.

3.2 Experimental Setting

To ensure experimental fairness and result stability, a fixed-length input sequence was generated for each dataset using a non-overlapping sliding window of length 100. The training data was split into training and validation sets with an 8:2 ratio. The experiments were conducted in Python on a system with a single Nvidia RTX 3090 GPU. For model configuration, the Transformer's hidden state channels (d_model) were set to 512, and the number of attention heads (h) was set to 8. The ADAM optimizer was used during training, with an initial learning rate of 5e^{-5}.

3.3 Comparison Methods

In the main experiment, we compared the METG model with 12 other models to evaluate its performance in a multivariate time series anomaly detection task. The experimental results are shown in Table 2. In this paper, we reproduce the results of Transformer, TimesNet[18] and Anomaly Transformer (A.T), while citing the performance results of other baselines in [15]. The baselines listed in Table 3 include classical machine learning techniques like LOF, OC-SVM and iForest[19], as well as deep learning models such as MPPCAD[20] and DAGMM[21] in density estimation models; clustering-based models, Deep-SVDD[22] and THOC[23]; Furthermore, the reconstruction-based models are also included, LSTM-VA[24], BeatGAN[25], Transformer, TimesNet, and A.T. The experimental results show that the average F1 scores of METG outperform the previous, more advanced models, A.T and TimesNet, demonstrating its excellent capability in multivariate time series anomaly detection.

3.4 Ablation Study

To validate the effectiveness of the key components of the proposed METG model, we conducted ablation experiments across four different datasets, evaluating model performance through F1 score comparisons. Table 3 shows in detail the performance of

Table 1: Four datasets details.

Datasets	Dim	Train	Valid	Test	P(%)
SMD	38	566724	141681	708420	0.5
MSL	55	46653	11664	73729	1
PSM	25	105984	26497	87841	1
SMAP	25	108146	27037	427617	1

Table 2: Results (%) and average F1 scores from four different domain datasets.

Datasets	**SMD**			**MSL**			**SMAP**			**PSM**			**Ave**
Metrics	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
LOF	56.34	39.86	46.68	57.89	90.49	70.61	57.60	72.15	65.43	72.15	65.43	68.42	62.79
OC-SVM	44.34	76.72	56.19	62.75	80.49	70.67	56.34	45.39	49.22	45.39	49.22	47.02	55.78
iForest	42.31	73.29	53.64	76.09	92.45	83.48	55.53	49.29	44.94	49.29	44.94	47.02	57.27
MMPCACD	71.20	79.28	75.02	76.25	77.29	75.74	81.73	82.52	68.29	82.52	68.29	74.73	73.45
DAGMM	67.30	49.89	57.30	75.84	95.41	70.18	68.51	89.92	57.42	89.92	57.42	68.92	63.46
DeepSVDD	78.54	79.67	79.10	95.41	86.49	80.97	69.04	80.42	90.84	80.42	90.84	85.12	84.01
THOC	79.76	90.95	84.99	88.14	89.73	88.18	90.68	83.94	85.13	83.94	85.13	88.88	86.80
LSTM-VAE	75.76	90.08	82.30	73.62	92.64	80.96	78.10	80.96	82.84	80.96	82.84	81.74	81.96
BeatGAN	84.06	79.07	81.49	74.86	90.30	83.94	69.61	92.04	93.72	92.04	93.72	87.84	86.75
Transformer	78.32	65.24	71.19	89.70	73.66	80.90	90.90	61.43	73.31	99.61	83.14	90.64	79.01
TimesNet	87.88	81.54	84.59	89.55	75.29	81.80	90.17	55.27	68.53	98.52	96.31	97.40	83.08
A.T	87.96	94.68	**91.20**	91.13	90.12	90.63	93.96	98.45	96.15	**96.81**	98.63	97.71	93.62
METG	**92.53**	**86.92**	89.64	**92.15**	**95.07**	**93.59**	**94.31**	**98.55**	**96.38**	93.62	**99.18**	**98.38**	**94.50**

Table 3: Results of ablation experiments on four datasets.

wo/M	wo/G	SMD	MSL	PSM	SMAP	Ave
×	×	72.86	82.66	78.94	69.99	76.11
×	√	89.60	91.86	91.47	71.76	86.17
√	×	88.15	87.65	96.51	70.49	85.70
√	√	89.64	93.59	98.38	96.38	94.50

METG and its different variants, where wo/M denotes the removal of the memory module and wo/G denotes the removal of the graph convolution module.

The experimental results demonstrate that the memory-enhanced module and GCN module are critical to METG's performance. Their combination enables more comprehensive and accurate anomaly detection. Removing the memory-enhanced module, the average F1 score decreases by 8.33%, especially by 24.62% on the SMAP dataset; Removing the GCN module, the average F1 score decreases by 8.8%, and the average F1 score on the SMAP dataset decreases to 70.49%. This suggests that the GCN module enhances the spatial modeling capability, thereby improving the accuracy of anomaly detection.

Additionally, experiments were conducted to evaluate the proposed anomaly detection criterion based on joint optimization. Table 4 compares METG's performance using different criteria, where L_1 stands for L_{MSE}, L$_2$ for L_{rec}, and M for m_{score}. The results show that performance is lower when using either a simple sum or a weighted product of these criteria. The best detection performance is achieved with our proposed criterion, which incorporates the importance of memory items in the overall query result.

3.5 Robustness analysis

To evaluate robustness, a key performance metric, the amount of training data is varied to assess the model's F1 score. As shown in Figure 3, increasing the proportion of the original training set used for training (20%, 40%, 60%, 80%, 100%) leads to a corresponding improvement in the F1 score. On the SMD, MSL, and PSM datasets, the model achieves strong F1 scores even with smaller training sets. The F1 values on the SMAP dataset show a linear increasing trend, and when the training data reaches 60%, the F1 values are all over 78%, which is better than the general algorithm. This indicates that the METG model can maintain good performance even when some of the training samples are reduced, showing good robustness.

Table 4: Ave F1 scores for different anomaly detection criteria.

Score	SMD	MSL	PSM	SMAP
L1+L2	76.21	88.24	79.04	69.39
M*L1+L2	77.27	88.20	82.73	69.24
M*L2+L1	76.87	89.28	84.55	71.49
M*(L1+L2)	89.64	93.59	98.38	96.38

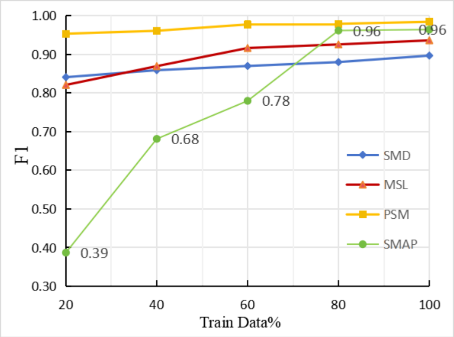

Figure 3: Robustness analysis.

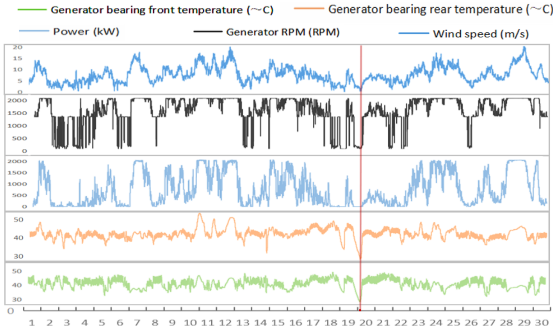

Figure 4: Visualization of generator data anomalies.

3.6 Case Study

Wind energy, a renewable and clean resource, offers great development potential. However, due to complex geographical factors, repairs following major failures of wind turbines are time-consuming and costly. As a key component, the reliability of the generator directly impacts the entire wind turbine's operation. Real-time health monitoring of the WTG generator using anomaly detection is essential. This study applies the METG model to a publicly available dataset[26] from the Penmanshiel wind farm in the UK, covering time-series data from January to June 2017 with 10-minute intervals. Using five generator parameters from WTG 1, the METG model achieves an accuracy of 0.9874 and an F1 score of 0.9921. Figure 4 shows the time-series curves of the generator attributes in June, with the red shading denoting the anomalous data detected, and the red line at the bottom denoting the anomalous time period marked in the test set. From the figure, it can be seen that the model is able to effectively detect WTG generator and converter error anomalies.

4 Conclusion

In this paper, we proposed a multivariate time series anomaly detection model METG based on joint training with unsupervised reconstruction. The model enhances the capability of capturing temporal features by introducing a memory-enhanced Transformer, which stores the different modes of normal data in the memory matrix. In addition, the model incorporates Graph Convolutional Network (GCN), which utilizes graph convolution operations to deeply extract features of data based on graph structures. Finally, the performance of METG is further optimized by co-training. Experimental results demonstrate that METG outperforms previous models, achieving outstanding detection effectiveness across several time series datasets from diverse domains.

References

[1] Ren H, Xu B, Wang Y, *et al.* Time-series anomaly detection service at Microsoft [C]// Proc of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 3009-3017.

[2] Gutoski M, Aquino N M R, Ribeiro M, *et al.* Detection of Video Anomalies Using Convolutional Autoencoders and One-Class Support Vector Machines [C]// CBIC ,2016:961-971.

[3] Hasan M, Choi J, Neumann J, *et al.* Learning temporal regularity in video sequences[C]//Proc of the IEEE conference on computer vision and pattern recognition. 2016: 733-742.

[4] Zhang Y, Wang J, Chen Y, *et al.* Adaptive Memory Networks With Self-Supervised Learning for Unsupervised Anomaly Detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2023(12):35.

[5] Zhang C, Song D, Chen Y, *et al.* A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data [C]// Proc of the AAAI. 2019, 33(01): 1409-1416.

[6] Xu H, Chen W, Zhao N, *et al.* Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications [C]// Proc of the 2018 world wide web conference. 2018:187-196.

[7] Li Z, Zhao Y, Han J, *et al.* Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding [C]// Proc of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2021: 3220-3230.

[8] Li D, Chen D, Jin B, *et al.* MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks [C]// International conference on artificial neural networks. 2019: 703-716.

[9] Abdulaal A, Liu Z, Lancewicki T. Practical approach to asynchronous multivariate time series anomaly detection and localization [C]// Proc of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2485-2494.

[10] Ding C, Sun S, Zhao J. MST-GAT: A multimodal spatial temporal graph attention network for time series anomaly detection[J]. Information Fusion, 2023, 89(2023): 527-536.

[11] Ashish V. Attention is all you need[J]. Advances in neural information processing systems,2017,30: I.

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

[13] Gong D, Liu L, Le V, *et al.* Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection [C]// Proc of the IEEE/CVF international conference on computer vision,. 2019: 1705-1714.

[14] Kipf T N, Welling M. Semi-Supervised Classifica-tion with Graph Convolutional Networks[J]. 2017.https://arxiv.org/abs/1609.02907

[15] Siffer A, Fouque P A, Termier A, *et al.* Anomaly detection in streams with extreme value theory [C]// Proc of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, New York: ACM Press. 2017: 1067-1075.

[16] Su Y, Zhao Y, Niu C, *et al.* Robust anomaly detection for multivariate time series through stochastic recurrent neural network [C]// Proc of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, New York: ACM Press. 2019: 2828-2837.

[17] Hundman K, Constantinou V, Laporte C, *et al.* Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding [C]// Proc of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, New York, ACM Press. 2018: 387-395.

[18] Wu H, Hu T, Liu Y, *et al.* Timesnet: Temporal 2d-variation modeling for general time series analysis[C]//The eleventh international conference on learning representations. 2022.

[19] Liu F T, Ting K M, Zhou Z H. Isolation forest [C]// 2008 eighth IEEE international conference on data mining. 2008: 413-422.

[20] Yairi T, Takeishi N, Oda T, *et al.* A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction[J]. IEEE Transactions on Aerospace and Electronic Systems, 2017, 53(3): 1384-1401.

[21] Zong B, Song Q, Min M R, *et al.* Deep autoencoding gaussian mixture ,model for unsupervised anomaly detection [C]// International conference on learning representations. 2018: 3072-3-91.

[22] Ruff L, Vandermeulen R, Goernitz N, *et al.* Deep one-class classification [C]//International conference on machine learning.2018: 4393-4402.

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

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

[25] Zhou B, Liu S, Hooi B, *et al.* Beatgan: Anomalous rhythm detection using adversarially generated time series [C]// Morgan Kaufmann: IJCAI. 2019: 4433-4439.

[26] Plumley C. Penmanshiel wind farm data[J]. Zenodo, doi, 2022, 10.