Dynamic graph neural networks and enhanced transformer for multivariate time series anomaly detection

Jiayi Li*[a], Jiali Liu[b], Xinwei Wang[b]
[a]China Academy of Industrial Internet, Beijing, China
[b]Beijing University of Posts and Telecommunications, State Key Lab of Networking and Switching Technology, Beijing, China
*Corresponding author: liijiayi@china-aii.com

ABSTRACT

Multivariate time series anomaly detection is crucial for ensuring the reliability of complex systems in industrial, healthcare, and IoT domains. However, existing methods struggle to effectively capture both dynamic inter-variable relationships and temporal dependencies. To address these challenges, we propose the Dynamic Graph Neural Networks and Enhanced Transformer (DGET) framework, which combines a Dynamic Graph Construction module to model evolving dependencies and an Enhanced Transformer architecture with dynamic temporal encoding, time-decay attention, and local window mechanisms for capturing temporal dependencies. Additionally, self-supervised contrastive learning enhances the model's robustness in label-scarce environments. We evaluate DGET on three benchmark datasets: SMD, SWaT, and WADI. Results show that DGET outperforms baseline methods (SVM, GNN, LSTM) and the state-of-the-art TranAD, achieving superior F1-scores. Ablation studies further confirm the critical roles of the Dynamic Graph Neural Networks and Enhanced Transformer components, demonstrating DGET's effectiveness for multivariate time series anomaly detection.

Keywords: Time Series, anomaly detection, GNN, transformer

1. INTRODUCTION

Multivariate time series anomaly detection is essential for ensuring reliability and efficiency in applications such as industrial monitoring, healthcare, and network security. However, it faces challenges in capturing complex static and dynamic dependencies, modeling both short-term and long-term temporal relationships, and addressing sparse and context-dependent anomalies. Traditional statistical methods like ARIMA[1], and machine learning approaches such as Isolation Forests[2] and one-class SVM[3] struggle with high-dimensional and nonlinear data. Deep learning models, including RNN[4], LSTM[5], and CNN[6], improve short-term dependency modeling but are limited in capturing long-term or global relationships. Transformer-based models[7] leverage self-attention for temporal modeling but suffer from high computational costs and sensitivity to noisy long-range dependencies, while GNN[8] effectively model variable relationships but lack dynamic dependency modeling.

To address these limitations, we propose the Dynamic Graph Neural Networks and Enhanced Transformer (DGET) framework, which combines Dynamic Graph Modeling for static and dynamic dependencies, an Enhanced Transformer for short-term and long-term temporal modeling, and Self-Supervised Contrastive Learning for improved robustness. Extensive experiments on benchmark datasets demonstrate the superiority of DGET over existing methods. The remainder of this paper discusses related work (Section 2), methodology (Section 3), experimental results (Section 4), and conclusions (Section 5).

2. RELATED WORK

Time series anomaly detection has been extensively studied, with various methods proposed to address the challenges in modeling temporal dependencies and inter-variable relationships in multivariate time series data.

Traditional methods such as ARIMA[1] and machine learning-based approaches like PCA[9] and Isolation Forest[2] have been applied for anomaly detection. However, these methods struggle with handling nonlinear temporal dependencies and high-dimensional data, limiting their applicability in complex real-world scenarios.

Second International Conference on Power Electronics and Artificial Intelligence (PEAI 2025), edited by Qiang Yang,
Parikshit N. Mahalle, Xuehe Wang, Proc. of SPIE Vol. 13657, 136573M
 · doi: 10.1117/12.3066832

Deep learning-based methods have shown significant progress in recent years. RNN[4] and LSTM[5] networks have been widely used to capture temporal dependencies. LSTM-based methods[5] effectively model short-term dependencies but often fail to capture long-term relationships due to their sequential nature. Kim et al. (2023)[10] proposed STOC, which combines stacked Transformer representations and a 1D convolutional decoder to fuse global and local features. This method achieved superior results on benchmark datasets; however, it lacks the capability to model inter-variable relationships explicitly.

To address the relationships between variables, GNN have been introduced. Tang et al. (2023)[11] proposed a method that integrates GNN with GRU to model multivariate relationships in industrial control systems, achieving improved interpretability and detection accuracy. However, the GRU structure introduces computational inefficiencies due to its sequential processing nature. Similarly, Belay et al. (2023)[12] explored GNN in their GDN framework to model static inter-variable relationships, but it does not account for dynamically evolving dependencies over time.

Transformer-based approaches have further advanced anomaly detection by leveraging self-attention mechanisms to capture global temporal dependencies. For example, Yang et al. (2023)[13] proposed DCdetector, a dual-attention contrastive learning model that amplifies differences between normal and abnormal representations. While DCdetector improves robustness using contrastive learning, it does not dynamically model inter-variable relationships, which are critical in multivariate time series. Belay et al. (2023)[12] also conducted a comprehensive comparative study on anomaly detection methods, highlighting the performance trade-offs between accuracy and computational efficiency in Transformer-based and graph-based approaches.

Despite these advancements, current methods either fail to integrate evolving inter-variable dependencies or struggle to balance short-term and long-term temporal dependency modeling. To address these limitations, we propose the Dynamic Graph Neural Networks and Enhanced Transformer (DGET) framework, which dynamically models evolving relationships between variables and effectively captures temporal dependencies through an enhanced Transformer architecture.

3. METHODOLOGY

In this section, we present the proposed Dynamic Graph Neural Networks and Enhanced Transformer (DGET) framework for multivariate time series anomaly detection. The framework is designed to capture evolving inter-variable relationships and effectively model both short-term and long-term temporal dependencies. The proposed DGET framework consists of four main modules. The overall structure of DGET is illustrated in Figure 1.

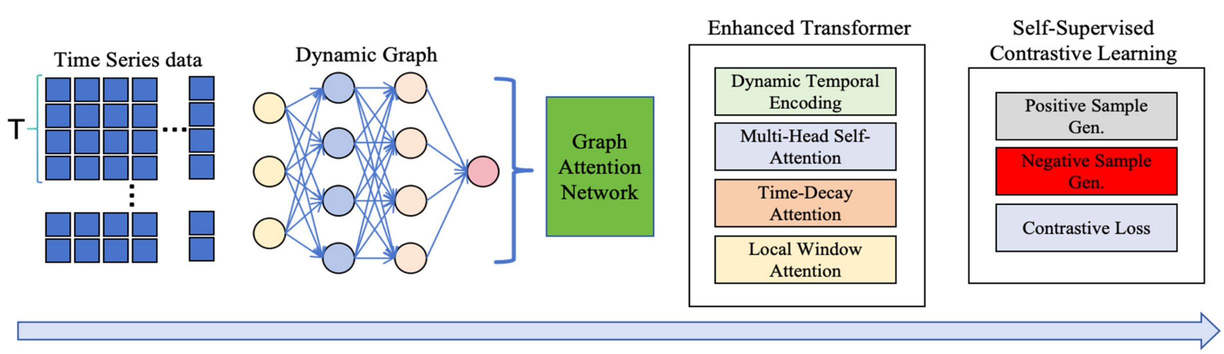

Figure 1. Overview of the DGET framework.

3.1 Dynamic graph construction

To model the evolving inter-variable dependencies in multivariate time series data, we construct a dynamic graph at each time step. Let $X_t = \{x_t^1, x_t^2, \cdots, x_t^N\}$ represent the multivariate input at time t,where N is the number of variables (e.g., sensors).

1. Graph Representation

Each variable x_t^i is treated as a node v_i, and edges e_{ij} between nodes represent the relationships among variables. A graph at time t can be defined as:

$$G_t = (V_t, E_t, A_t) \tag{1}$$

where V_t is the set of nodes, E_t represents the edges, and $A_t \in \mathbb{R}^{N \times N}$ is the adjacency matrix.

2. Dynamic Edge Weights

The edge weights $A_t(i,j)$ between nodes v_i and v_j are dynamically updated based on a sliding window of past observations W:

$$A_t(i,j) = \mathrm{sim}\left(x_t^i, x_t^j\right) \tag{2}$$

where $\mathrm{sim}(\cdot,\cdot)$ computes the correlation or distance between variables.

3. Graph Dynamics

By sliding the window over the time series, the graph G_t evolves over time, allowing the model to adapt to changing relationships.

3.2 Graph attention-based embedding

The dynamic graph G_t is passed to a GNN to learn node embeddings that encode both static and dynamic inter-variable relationships.

1. Graph Attention Network (GAT)

We employ a Graph Attention Network (GAT) to assign varying importance to different edges using an attention mechanism. For a node v_i, the attention coefficients α_{ij} are computed as:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^T[\mathbb{W}h_i \| \mathbb{W}h_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(\mathrm{LeakyRelu}\left(a^T[\mathbb{W}h_i \| \mathbb{W}h_k]\right)\right)} \tag{3}$$

Where h_i and h_j are the features of nodes v_i and v_j, a is the attention weight vector, and $\mathbb{W}$ is a learnable transformation matrix.

2. Output Node Embeddings

The output node embeddings $H_t = \{h_t^1, h_t^2, \cdots, h_t^N\}$ capture the dynamic relationships and are passed as inputs to the next module.

3.3 Enhanced transformer-based temporal modeling

To capture both short-term and long-term temporal dependencies, we introduce an Enhanced Transformer architecture with the following improvements:

1. Dynamic Temporal Encoding

Unlike fixed positional encodings, a learnable dynamic temporal encoding is used:

$$TE(t) = \sin(\omega t) + \mathbb{W}_{\mathrm{dynamic}} \cdot X_t \tag{4}$$

where ω is a frequency parameter, and $\mathbb{W}_{\mathrm{dynamic}}$ is a learnable transformation matrix.

2. Time-Decay Attention

To reduce the influence of irrelevant long-term dependencies, a decay function is applied to the attention weights:

$$A_{ij} = \frac{\exp\left(\frac{Q_i K_j^T}{\sqrt{d_k}} - \lambda|t_i - t_j|\right)}{\sum_k \exp\left(\frac{Q_i K_k^T}{\sqrt{d_k}} - \lambda|t_i - t_k|\right)} \tag{5}$$

where λ is a decay factor, and Q, K are query and key projections.

3. Local Window Attention

Self-attention is restricted to a local window size L around each time step, reducing computational complexity to $O(TL)$ while retaining critical temporal patterns.

The output of the Enhanced Transformer is a set of temporal embeddings Z_t, which encode short-term and long-term dependencies.

3.4 Anomaly detection with self-supervised contrastive learning

To enhance robustness in distinguishing anomalies, a self-supervised contrastive learning strategy is employed:

1. Positive and Negative Samples:

Positive samples are generated by slightly augmenting normal sequences, while negative samples are obtained using corrupted or time-shifted versions of the sequences.

2. Contrastive Loss:

A contrastive objective is used to maximize the similarity between positive pairs and minimize the similarity with negative samples:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{6}$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity and τ is a temperature scaling factor.

3. Anomaly Score Computation:

The anomaly score S_t is computed as:

$$S_t = \left\| X_t - \hat{X}_t \right\|_2 + \text{sim}(z_t, z_{\text{ref}}) \tag{7}$$

where $\hat{X}_t$ is the reconstructed sequence and z_{ref} is the reference normal embedding.

4. EVALUATION

In this section, we conduct extensive experiments to evaluate the performance of the proposed Dynamic Graph Neural Networks and Enhanced Transformer (DGET) framework on three publicly available multivariate time series anomaly detection datasets: SMD, SWaT, and WADI. The evaluation includes comparative experiments against baseline and state-of-the-art (SOTA) methods, as well as ablation studies to validate the contributions of each key component in the proposed model. Precision, Recall, and F1-score are used as the evaluation metrics.

4.1 Datesets

We evaluate DGET on the following three widely used benchmark datasets:

➓ SMD (Server Machine Dataset): A multivariate dataset containing sensor measurements from 28 servers. It is commonly used to benchmark anomaly detection methods in high-dimensional and noisy industrial data.

➓ SWaT (Secure Water Treatment): A dataset collected from a water treatment testbed consisting of 51 sensors. It simulates an industrial control system with labeled normal and abnormal behaviors.

➓ WADI (Water Distribution): A dataset collected from a water distribution system with 123 sensors. It represents realistic industrial scenarios with labeled anomalies.

Each dataset contains labeled timestamps that distinguish between normal and anomalous events, allowing for supervised performance evaluation.

4.2 Evaluation metrics

We use precision, recall and F1score to evaluate the detection performance of all models.

4.3 Comparative experiments

To evaluate the effectiveness of the proposed DGET framework, we compared its performance with several baseline and state-of-the-art (SOTA) approaches on the SMD, SWaT, and WADI datasets. The baseline methods include SVM, GNN, LSTM. For the SOTA comparison, we used TranAD[14], a Transformer-based anomaly detection model that leverages self-attention mechanisms to model temporal dependencies.

The experimental results, presented in Table 1, show that DGET achieves the highest F1-scores across all datasets, outperforming all baseline methods as well as the SOTA method TranAD. These results demonstrate the effectiveness of the proposed framework in combining dynamic graph modeling and an enhanced Transformer architecture to capture both inter-variable and temporal dependencies. DGET's superior performance highlights its ability to address the limitations of traditional baselines and existing SOTA methods, making it a robust solution for multivariate time series anomaly detection.

Table 1. Performance comparison of DGET with other methods on the complete dataset.

Datasets	Methods	Precision	Recall	F1score
SMD	SVM	0.8022	0.8419	0.8216
	GNN	0.8526	0.8723	0.8623
	LSTM	0.8993	0.9001	0.8997
	TranAD	0.9262	0.9974	0.9605
	DGET	**0.9307**	**0.9936**	**0.9611**
SWaT	SVM	0.8430	0.5077	0.6337
	GNN	0.8264	0.5221	0.6399
	LSTM	0.9037	0.5530	0.6861
	TranAD	0.9760	0.6997	0.8151
	DGET	**0.9792**	**0.7266**	**0.8342**
WADI	SVM	0.1969	0.6323	0.3003
	GNN	0.2688	0.6095	0.3731
	LSTM	0.2365	0.6927	0.3526
	TranAD	0.3529	0.8296	0.4951
	DGET	**0.3771**	**0.8693**	**0.5260**

4.4 Ablation study

To validate the contributions of the Dynamic Graph Neural Networks (DGN) and Enhanced Transformer (ET) components in DGET, we conduct ablation studies by removing each component from the proposed model:

❿ DGET w/o DGN: The dynamic graph module is removed, and the model uses only the enhanced Transformer for temporal dependency modeling.

❿ DGET w/o ET: The enhanced Transformer is replaced with a simple LSTM model while retaining the dynamic graph component.

The ablation study results on SMD, SWaT, and WADI datasets are shown in Table 2.

Table 2. Ablation study – F1score for DGET and its ablated versions.

Model	SMD(F1score)	SWaT(F1score)	WADI(F1score)
DGET w/o DGN	0.8271	0.6233	0.2872
DGET w/o ET	0.8503	0.6082	0.3325
DGET (Full)	**0.9611**	**0.8342**	**0.5260**

Through the above experiments, it can be observed that removing the Dynamic Graph Neural Networks (DGN) module significantly reduces performance, confirming its critical role in capturing dynamic inter-variable relationships. Similarly, replacing the Enhanced Transformer with LSTM results in lower F1-scores, highlighting the Enhanced Transformer’s ability to effectively model both short-term and long-term temporal dependencies. The full DGET model achieves the best results, demonstrating the complementary and indispensable roles of both the DGN and Enhanced Transformer in enhancing anomaly detection performance.

5. CONCLUSION

In this paper, we proposed the Dynamic Graph Neural Networks and Enhanced Transformer (DGET) framework for multivariate time series anomaly detection. DGET effectively models evolving inter-variable relationships using dynamic graph construction and graph attention networks, while the Enhanced Transformer captures both short-term and long-term temporal dependencies with dynamic temporal encoding, time-decay attention, and local window mechanisms.

Experiments on benchmark datasets demonstrated that DGET outperforms baseline and state-of-the-art methods, achieving superior F1-scores. Ablation studies confirmed the complementary roles of its key components.

Despite its strengths, DGET has limitations that provide directions for future work. One limitation is the computational complexity associated with dynamic graph updates and Transformer operations, which may impact its suitability for real-time or large-scale applications. Future efforts could focus on optimizing these processes to enhance efficiency. Additionally, extending the framework to handle irregularly-sampled time series and more diverse anomaly types, such as collective anomalies, would broaden its applicability. Addressing these challenges will further solidify DGET's robustness and impact in multivariate time series anomaly detection.

REFERENCES

[1] Kontopoulou, V. I., Panagopoulos, A. D., Kakkos, I., & Matsopoulos, G. K. (2023). A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. Future Internet, 15(8), 255.

[2] Xu, H., Pang, G., Wang, Y., & Wang, Y. (2023). Deep isolation forest for anomaly detection. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12591-12604.

[3] Pinon, N., Trombetta, R., & Lartizien, C. (2024, January). One-Class SVM on siamese neural network latent space for Unsupervised Anomaly Detection on brain MRI White Matter Hyperintensities. In Medical Imaging with Deep Learning (pp. 1783-1797). PMLR.

[4] Ullah, I., & Mahmoud, Q. H. (2022). Design and development of RNN anomaly detection model for IoT networks. IEEE Access, 10, 62722-62750.

[5] Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. Computers in Industry, 131, 103498.

[6] Ullah, W., Hussain, T., Ullah, F. U. M., Lee, M. Y., & Baik, S. W. (2023). TransCNN: Hybrid CNN and transformer mechanism for surveillance anomaly detection. Engineering Applications of Artificial Intelligence, 123, 106173.

[7] Ashish, V. (2017). Attention is all you need. Advances in neural information processing systems, 30, I.

[8] Wang, Y., Zhang, J., Guo, S., Yin, H., Li, C., & Chen, H. (2021, July). Decoupling representation learning and classification for gnn-based anomaly detection. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval (pp. 1239-1248).

[9] Hoang, D. H., & Nguyen, H. D. (2018, February). A PCA-based method for IoT network traffic anomaly detection. In 2018 20th International conference on advanced communication technology (ICACT) (pp. 381-386). IEEE.

[10] Kim, J., Kang, H., & Kang, P. (2023). Time-series anomaly detection with stacked Transformer representations and 1D convolutional network. Engineering Applications of Artificial Intelligence, 120, 105964.

[11] Tang, C., Xu, L., Yang, B., Tang, Y., & Zhao, D. (2023). GRU-based interpretable multivariate time series anomaly detection in industrial control system. Computers & Security, 127, 103094.

[12] Belay, M. A., Blakseth, S. S., Rasheed, A., & Salvo Rossi, P. (2023). Unsupervised anomaly detection for IoT-based multivariate time series: Existing solutions, performance analysis and future directions. Sensors, 23(5), 2844.

[13] Yang, Y., Zhang, C., Zhou, T., Wen, Q., & Sun, L. (2023, August). Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3033-3045).

[14] Tuli, S., Casale, G., & Jennings, N. R. (2022). Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.